Bridging Deep Neural Networks and Differential Equations for Image Analysis and Beyond

Bin Dong (董彬)

- Beijing International Center for Mathematical Research, Peking University
- Laboratory for Biomedical Image Analysis & Laboratory of Deep Learning Research, Beijing Institute of Big Data Research

Outline

- Deep "revolution" in imaging science
- Bridging Deep Neural Networks (DNNs) and Differential Equations
  - Dynamics perspective on deep learning for image classification
  - PDE-Net: learning PDEs from data
  - Moving endpoint control for blind image recovery

“Deep Revolution” in Imaging Science

Push learning to the extreme

30 Years of Image Restoration

- Image restoration: $f = Au + \eta$
- Variational and Optimization Models

$$\min_{u} \lambda R(u) + \|Au - f\|^2$$

  - Total variation (TV) and generalizations: $R(u) = \|\nabla u\|_1 \quad \text{or} \quad \|\boldsymbol{D}u\|_1$
  - Wavelet frame based: $R(u) = \|Wu\|_1 \quad \text{or} \quad \|Wu\|_0$
  - Others: total generalized variation, low rank, NLM, BM3D, K-SVD, data-driven tight frame, etc.
- PDEs and Iterative Algorithms
  - Perona-Malik equation, shock-filtering (Rudin & Osher), etc

$$u_t = \sum_{\ell=1}^{L} \frac{\partial^{\boldsymbol{\alpha}_\ell}}{\partial x^{\boldsymbol{\alpha}_\ell}} \Phi_\ell(\boldsymbol{D}u, u) - A^*(Au - f), \qquad \text{with } \boldsymbol{D} = \left(\frac{\partial^{\boldsymbol{\beta}_1}}{\partial x^{\boldsymbol{\beta}_1}}, \ldots, \frac{\partial^{\boldsymbol{\beta}_L}}{\partial x^{\boldsymbol{\beta}_L}}\right)$$

  - Iterative shrinkage algorithm

$$\boldsymbol{u}^k = \widetilde{\boldsymbol{W}}^\top \boldsymbol{S}_{\boldsymbol{\alpha}^{k-1}}(\boldsymbol{W}\boldsymbol{u}^{k-1}) - \boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{u}^{k-1} - \boldsymbol{f}), \quad k = 1, 2, \cdots$$

“Deep Revolution” in Imaging Science

- Development:
  - Handcraft modeling (1990-)
    - Variational and PDE Models: total variation, Perona-Malik, shock-filters, nonlocal, etc.
    - Applied harmonic analysis: wavelets, wavelet frames, etc.
  - Handcraft + data driven modeling (1999-):
    - Linear representation learning: MOD, K-SVD, data-driven tight frame, Ada-frame, low rank, etc.
    - Statistical models
  - Deep learning (2012-): CNNs/RNNs

More data and better processors

Deep Learning

- What are still challenging
  - Learning from limited or/and weakly labelled data
  - Learning from data of different types
  - Theoretical guidance, transparency
- How to provide guidance and transparency to deep learning?

Find “frameworks” and “links” with mathematics.

Deep Network	⟺	Differential Equations (DE)
Network Architecture	⟺	Numerical DE Our contribution
Network Training	⟺	Optimal Control

Weinan. E. Communications in Mathematics and Statistics, 5(1):1–11, 2017.

Brings new insights and applications

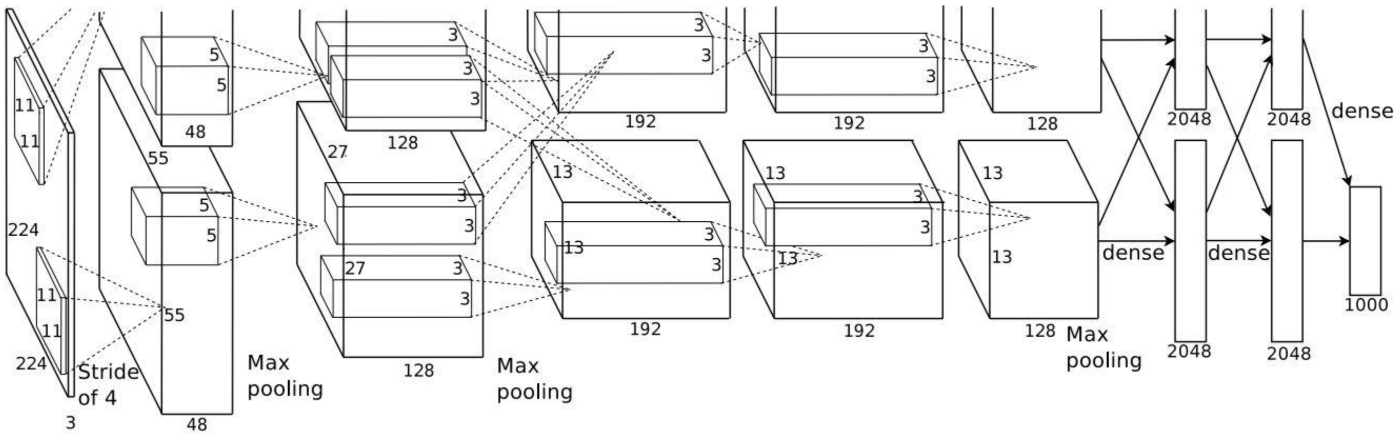
Bridging Differential Equations with Deep Networks

DNNs and numerical ODEs

- Yiping Lu, Aoxiao Zhong, Quanzheng Li and Bin Dong, *Beyond Finite Layer Neural Networks: Bridging Deep Architectures and Numerical Differential Equations*, ICML 2018. (arXiv:1710.10121)

DEEP NEURAL NETWORKS

AlexNet: *[Krizhevsky et al. 2012]*



Deep Neural Network
$f_1(f_2(f_3 \cdots (x)))$

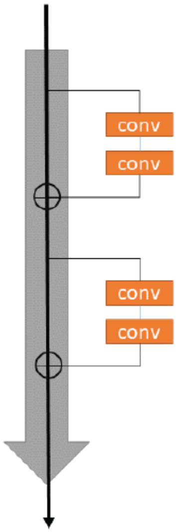
A Dynamic System?

Bridging Deep Architectures and Numerical Differential Equations

- Residual networks as discretizations of dynamic systems

Discrete: $$u_{n+1} = u_n + \Delta t \cdot f(u_n, t_n)$$

Continuum: $$u_t = f(u, t)$$

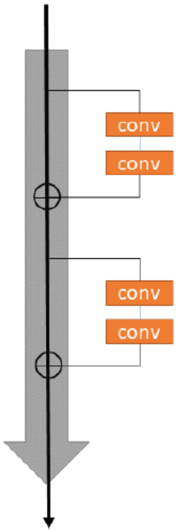


ResNet:

- K. He et al., Deep residual learning for image recognition. CVPR 2015.
- K. He et al., Identity mappings in deep residual networks. CVPR 2016.

BRIDGING DEEP ARCHITECTURES AND NUMERICAL DIFFERENTIAL EQUATIONS

- Residual networks as discretizations of dynamic systems



  Related work: ResNet and dynamic system

  - W. E. Communications in Mathematics and Statistics, 5(1):1–11, 2017.
  - S. Sonoda and N. Murata. ICML Workshop 2017
  - Z. Li and Z. Shi. arXiv preprint arXiv:1708.05115, 2017.
  - B. Chang, et al. arXiv preprint arXiv:1709.03698, 2017.

- Remaining questions:
  - Can we relate more networks with dynamic systems?
  - What can we do with such observation?

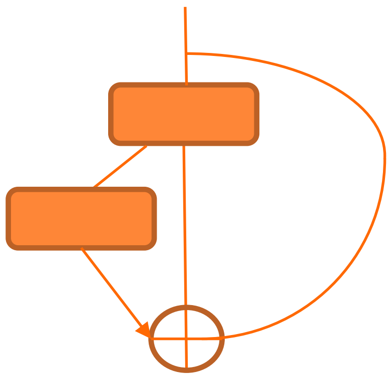
ResNet:

- K. He et al., Deep residual learning for image recognition. CVPR 2015.
- K. He et al., Identity mappings in deep residual networks. CVPR 2016.

Bridging Deep Architectures and Numerical Differential Equations

- Residual networks as discretizations of dynamic systems

$$x_{n+1} = x_n + F(x_n) + F(F(x_n))$$



PolyNet

Backward Euler Scheme:

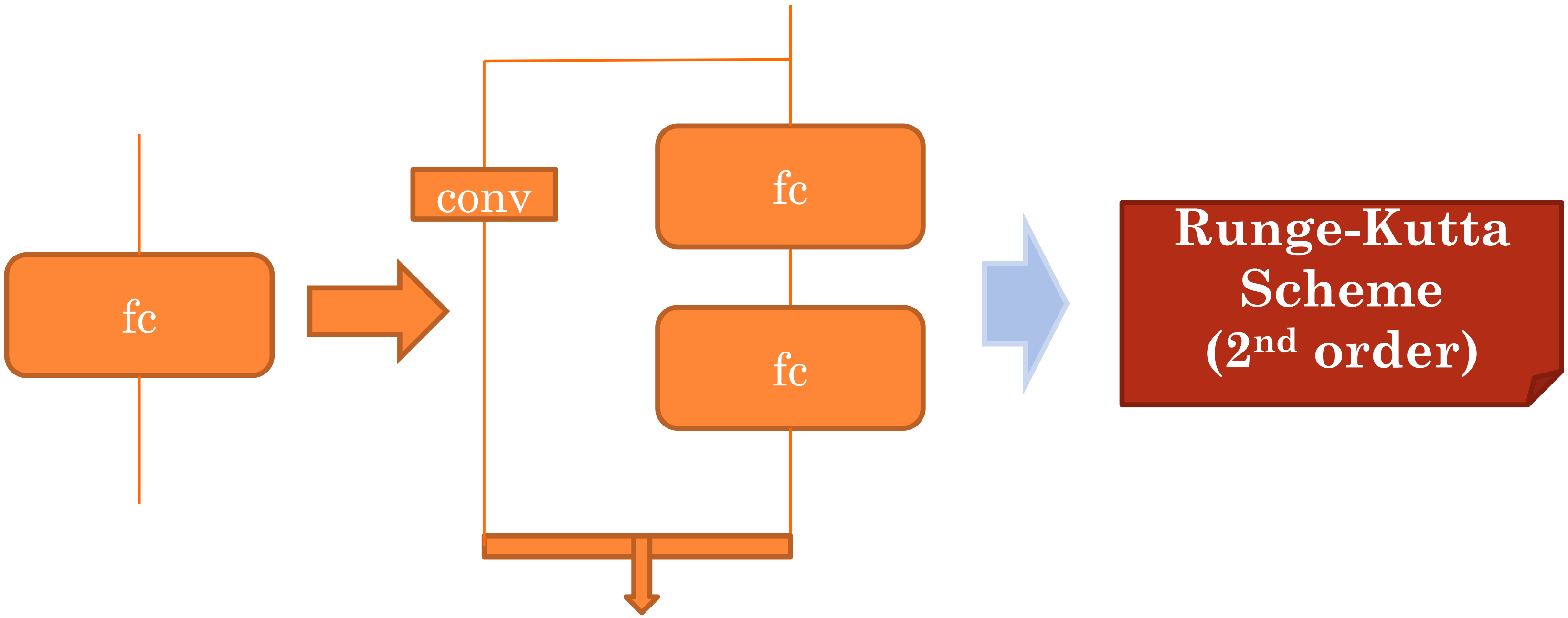
$$x_{n+1} = x_n + F(x_{n+1}) \Rightarrow x_{n+1} = (I - F)^{-1} x_n$$

Approximate the operator $(I - F)^{-1}$ by $I + F + F^2 + \cdots$

Zhang X, et al. PolyNet: A Pursuit of Structural Diversity in Very Deep Networks, CVPR 2017.

Bridging Deep Architectures and Numerical Differential Equations

- Residual networks as discretizations of dynamic systems



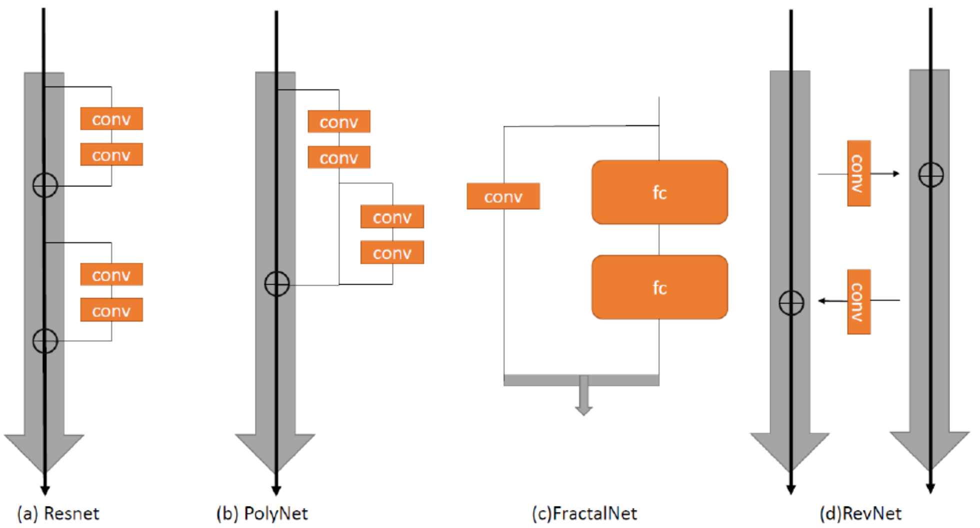
$$x_{n+1} = k_1 x_n + k_2(k_3 x_n + f_1(x_n)) + f_2(k_3 x_n + f_1(x_n))$$

Larsson G, Maire M, Shakhnarovich G. FractalNet: Ultra-Deep Neural Networks without Residuals, ICLR 2017.

Bridging Deep Architectures and Numerical Differential Equations

- Residual networks as discretizations of dynamic systems

Popular deep residual networks

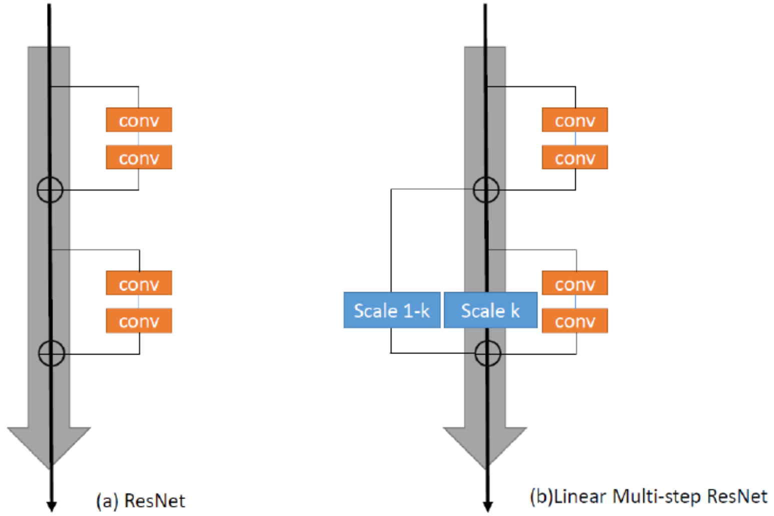


Network	Related ODE	Numerical Scheme
ResNet, ResNeXt, etc.	$u_t = f(u)$	Forward Euler scheme
PolyNet	$u_t = f(u)$	Approximation of backward Euler scheme
FractalNet	$u_t = f(u)$	Runge-Kutta scheme
Revnet	$\dot{X} = f_1(Y), \dot{Y} = f_2(X)$	Forward Euler scheme

Bridging Deep Architectures and Numerical Differential Equations

- Numerical differential equation inspired networks
  - Linear multi-step structure (LM-structure)

$$u_{n+1} = (1 - k_n)u_n + k_n u_{n-1} + \Delta t \cdot f(u_n, t_n)$$



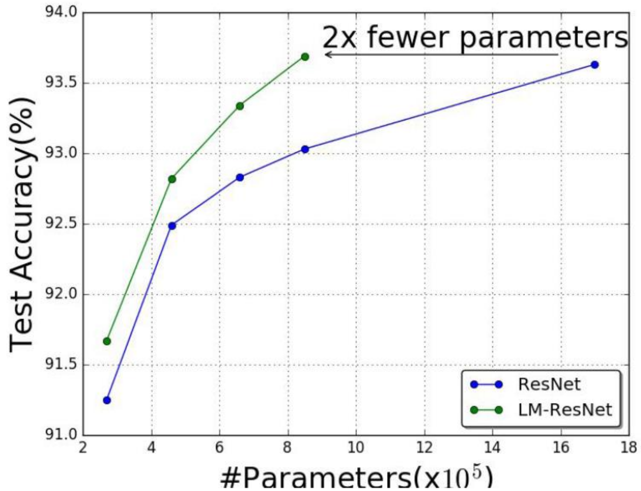
  - Can be applied to any ResNet-like networks. Examples: *LM-ResNet* and *LM-ResNeXt*

ResNeXt: Xie et al. Aggregated residual transformations for deep neural networks. CVPR 2017.

Bridging Deep Architectures and Numerical Differential Equations

- Results: CIFAR10 (50k train, 10k test, 10 classes)

Model	Layer	Error	Params
ResNet (He et al. (2015b))	20	8.75	0.27M
ResNet (He et al. (2015b))	32	7.51	0.46M
ResNet (He et al. (2015b))	44	7.17	0.66M
ResNet (He et al. (2015b))	56	6.97	0.85M
ResNet (He et al. (2016))	110, pre-act	6.37	1.7M
LM-ResNet (Ours)	20, pre-act	8.33	0.27M
LM-ResNet (Ours)	32, pre-act	7.18	0.46M
LM-ResNet (Ours)	44, pre-act	6.66	0.66M
LM-ResNet (Ours)	56, pre-act	**6.31**	0.85M



Modified equation perspective:

ResNet:

$$\dot{u}_n + \frac{\Delta t}{2}\ddot{u}_n = f(u_n)$$

LM-ResNet:

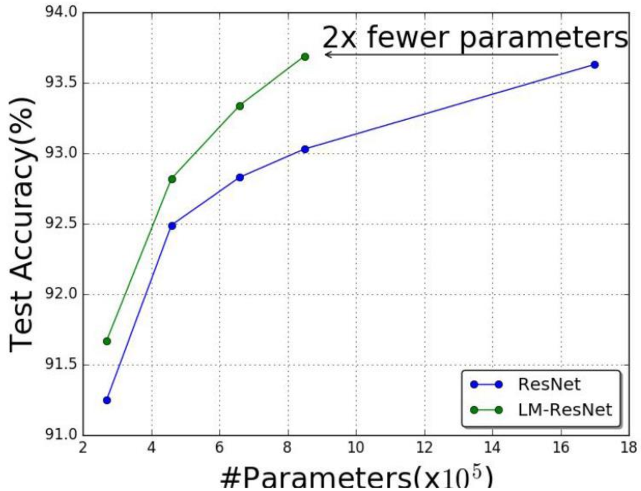
$$(1+k_n)\dot{u}_n + (1-k_n)\frac{\Delta t}{2}\ddot{u}_n = f(u_n)$$

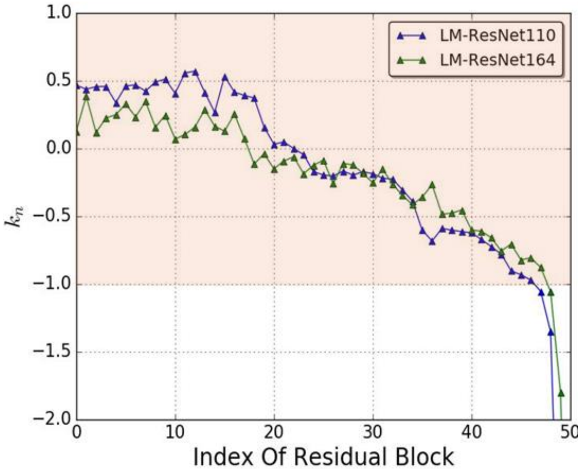
Speeds up and stable in $(-1, 0)$

BRIDGING DEEP ARCHITECTURES AND NUMERICAL DIFFERENTIAL EQUATIONS

- Results: CIFAR10 (50k train, 10k test, 10 classes)

Model	Layer	Error	Params
ResNet (He et al. (2015b))	20	8.75	0.27M
ResNet (He et al. (2015b))	32	7.51	0.46M
ResNet (He et al. (2015b))	44	7.17	0.66M
ResNet (He et al. (2015b))	56	6.97	0.85M
ResNet (He et al. (2016))	110, pre-act	6.37	1.7M
LM-ResNet (Ours)	20, pre-act	8.33	0.27M
LM-ResNet (Ours)	32, pre-act	7.18	0.46M
LM-ResNet (Ours)	44, pre-act	6.66	0.66M
LM-ResNet (Ours)	56, pre-act	**6.31**	0.85M





Bridging Deep Architectures and Numerical Differential Equations

- Results: CIFAR10 (50k train, 10k test, 10 classes)

Model	Layer	Error	Params
ResNet (He et al. (2015b))	20	8.75	0.27M
ResNet (He et al. (2015b))	32	7.51	0.46M
ResNet (He et al. (2015b))	44	7.17	0.66M
ResNet (He et al. (2015b))	56	6.97	0.85M
ResNet (He et al. (2016))	110, pre-act	6.37	1.7M
LM-ResNet (Ours)	20, pre-act	8.33	0.27M
LM-ResNet (Ours)	32, pre-act	7.18	0.46M
LM-ResNet (Ours)	44, pre-act	6.66	0.66M
LM-ResNet (Ours)	56, pre-act	**6.31**	0.85M

- Original motivation:
  - Nesterov ODE "Su, Boyd and Candes, NIPS 2014"; "Wibisono, Wilson and Jordan, PNAS 2016"
  - Nesterov PDE "Dong, Jiang and Shen, MMS, 2017 (UCLA CAM Report, Dec. 2013)"

BRIDGING DEEP ARCHITECTURES AND NUMERICAL DIFFERENTIAL EQUATIONS

- Results: CIFAR100 (50k train, 10k test, 100 classes)

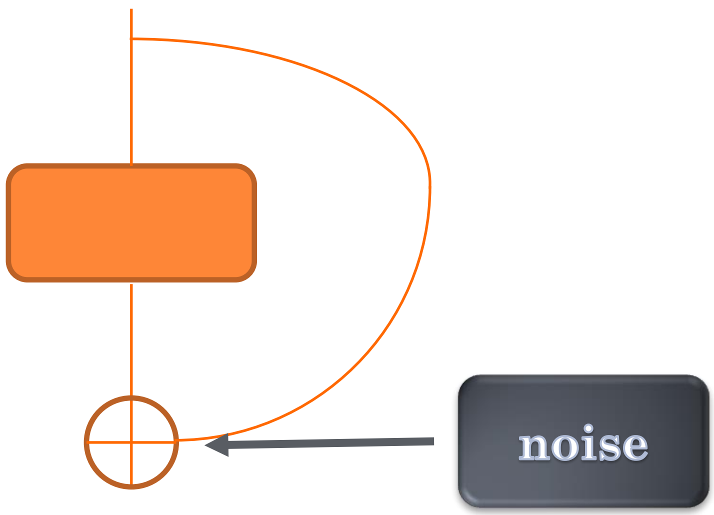
ResNet (Huang et al. (2016b))	110, pre-act	27.76	1.7M	CIFAR100
ResNet (He et al. (2016))	164, pre-act	24.33	2.55M	CIFAR100
ResNet (He et al. (2016))	1001, pre-act	22.71	18.88M	CIFAR100
FractalNet (Larsson et al. (2016))	20	23.30	38.6M	CIFAR100
FractalNet (Larsson et al. (2016))	40	22.49	22.9M	CIFAR100
DenseNet (Huang et al., 2016a)	100	19.25	27.2M	CIFAR100
DenseNet-BC (Huang et al., 2016a)	190	17.18	25.6M	CIFAR100
ResNeXt (Xie et al. (2017))	29(8×64d)	17.77	34.4M	CIFAR100
ResNeXt (Xie et al. (2017))	29(16×64d)	17.31	68.1M	CIFAR100
ResNeXt (Our Implement)	29(16×64d), pre-act	17.65	68.1M	CIFAR100
LM-ResNet (Ours)	110, pre-act	25.87	1.7M	CIFAR100
LM-ResNet (Ours)	164, pre-act	22.90	2.55M	CIFAR100
LM-ResNeXt (Ours)	29(8×64d), pre-act	17.49	34.4M	CIFAR100
LM-ResNeXt (Ours)	29(16×64d), pre-act	**16.79**	68.1M	CIFAR100

- Results: ImageNet (1.28m train, 50k test, 1000 classes)

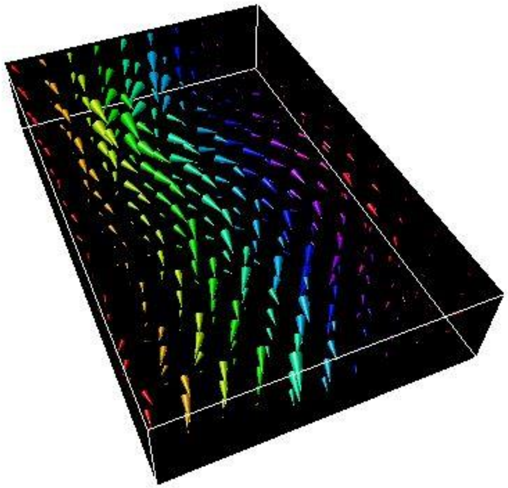
Model	Layer	top-1	top-5
ResNet (He et al. (2015b))	50	24.7	7.8
ResNet (He et al. (2015b))	101	23.6	7.1
ResNet (He et al. (2015b))	152	23.0	6.7
LM-ResNet (Ours)	50, pre-act	23.8	7.0
LM-ResNet (Ours)	101, pre-act	**22.6**	**6.4**

Bridging Deep Architectures and Numerical Differential Equations

- Stochastic training strategy



Dynamic System



Better generalization

Bridging Deep Architectures and Numerical Differential Equations

- Stochastic training strategy as stochastic control

$$\min \mathbb{E}_{X(0)\sim data}\left(\mathbb{E}(L(X(T))+\int_0^T R(\theta))\right)$$
$$s.t.\ dX=f(X,\theta)dt+g(X,\theta)dB_t$$

  - Examples (through weak convergence):
    - shake-shake regularization (Gastaldi, ICLR Workshop 2017)
    - stochastic depth (Huang et al. ECCV 2016)
- LM-structure + stochastic training

$$X_{n+1}=(2+g_n)X_n-(1+g_n)X_{n-1}+\eta_n f(X_n)$$

  - Distribution of η_n
    - Gaussian
    - Bernoulli

Bridging Deep Architectures and Numerical Differential Equations

- Results: $\eta_n \sim$ Bernoulli

Model	Layer	Training Strategy	Error
ResNet(He et al. (2015b))	110	Original	6.61
ResNet(He et al. (2016))	110,pre-act	Orignial	6.37
ResNet(Huang et al. (2016b))	56	Stochastic depth	5.66
ResNet(Our Implement)	56,pre-act	Stochastic depth	5.55
ResNet(Huang et al. (2016b))	110	Stochastic depth	5.25
ResNet(Huang et al. (2016b))	1202	Stochastic depth	4.91
LM-ResNet(Ours)	56,pre-act	Stochastic depth	5.14
LM-ResNet(Ours)	110,pre-act	Stochastic depth	**4.80**

Bridging Deep Architectures and Numerical Differential Equations

- Summary:
  - Bridge numerical differential equations with deep neural networks
  - Bridge numerical stochastic differential equations with stochastic training strategies
  - This new perspective inspired new network design (LM-architecture) that can reduce 40%~90% of the parameters of some deep networks with comparable accuracy
  - Such performance boost can be explained using modified equations

Other Related Works

- **Architecture design**
  - Haber E, Ruthotto L. Stable architectures for deep neural networks. Inverse Problems, 2017, 34(1): 014004.
  - Chang B, Meng L, Haber E, et al. Reversible architectures for arbitrarily deep residual neural networks. AAAI 2018.
  - Chang B, Meng L, Haber E, et al. Multi-level residual networks from dynamical systems view. ICLR 2018.
  - Wang B, Yuan B, Shi Z, et al. EnResNet: ResNet Ensemble via the Feynman-Kac Formalism. arXiv:1811.10745, 2018.
  - Tao Y, Sun Q, Du Q, et al. Nonlocal Neural Networks, Nonlocal Diffusion and Nonlocal Modeling. NeurIPS 2018.
  - Zhu M, Chang B, Fu C. Convolutional Neural Networks combined with Runge-Kutta Methods. arXiv:1802.08831, 2018.
  - Zhang L, Schaeffer H. Forward Stability of ResNet and Its Variants. arXiv:1811.09885, 2018.
  - Sun Q, Tao Y, Du Q. Stochastic Training of Residual Networks: a Differential Equation Viewpoint. arXiv:1812.00174, 2018.
  - He J, Xu J. MgNet: A Unified Framework of Multigrid and Convolutional Neural Network. arXiv:1901.10415, 2019.
  - Zhang J, Han B, Wynter L, et al. Towards Robust ResNet: A Small Step but A Giant Leap. arXiv:1902.10887, 2019.

Other Related Works

- **Optimization**
  - Li Q, Chen L, Tai C, E W. Maximum principle based algorithms for deep learning. The Journal of Machine Learning Research, 2017, 18(1): 5998-6026.
  - Li Q, Hao S. An optimal control approach to deep learning and applications to discrete-weight neural networks. ICML 2018.
  - Chen T Q, Rubanova Y, Bettencourt J, et al. Neural ordinary differential equations. NeurIPS 2018. (Best paper)
  - Parpas P, Muir C. Predict Globally, Correct Locally: Parallel-in-Time Optimal Control of Neural Networks. arXiv:1902.02542.
- **Theory**
  - E W., Han J, Li Q. A mean-field optimal control formulation of deep learning. Research in the Mathematical Sciences, vol. 6, no. 10, pp. 1–41, 2019.
  - Thorpe M, van Gennip Y. Deep Limits of Residual Neural Networks. arXiv:1810.11741, 2018.

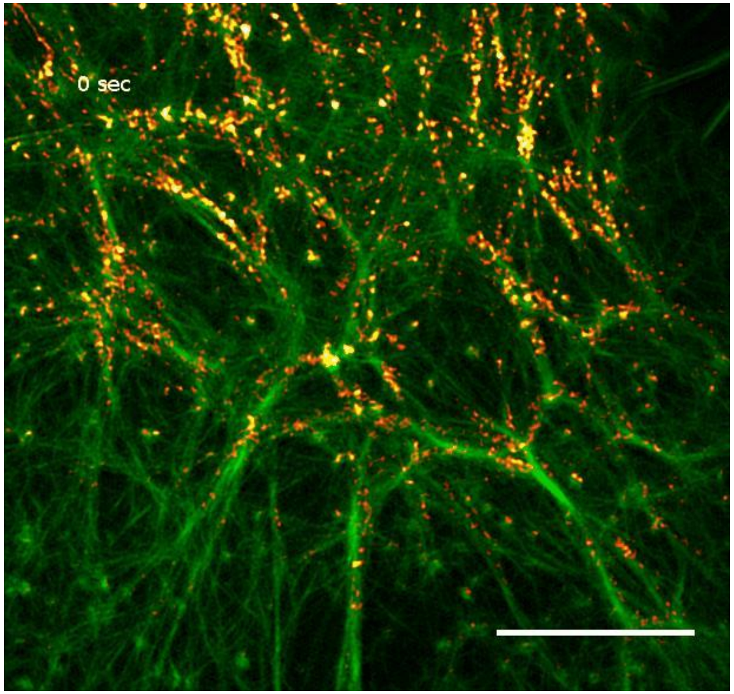
Bridging Differential Equations with Deep Networks

DNNs and numerical PDEs

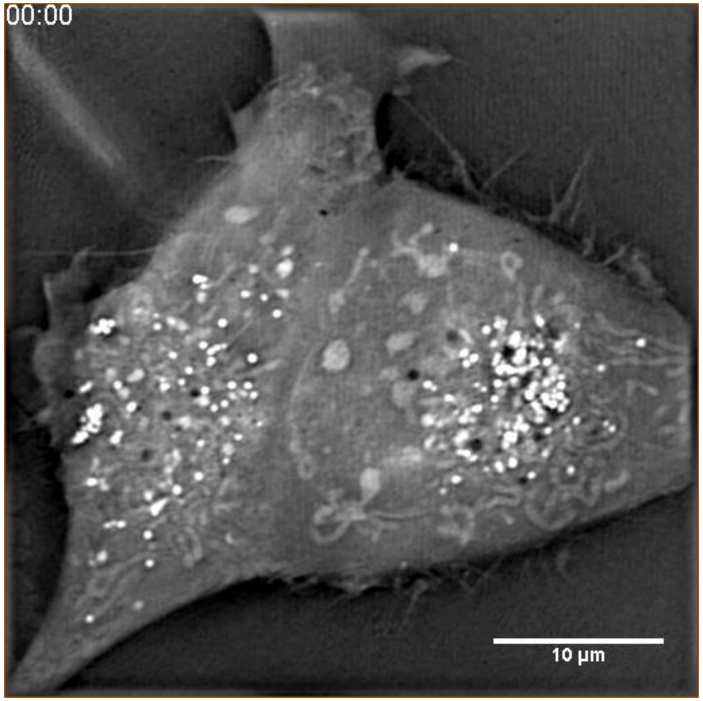
- Zichao Long, Yiping Lu, Xianzhong Ma and Bin Dong, *PDE-Net: Learning PDEs from Data*, ICML 2018. (arXiv:1710.09668)
- Zichao Long, Yiping Lu and Bin Dong, *PDE-Net 2.0: Learning PDEs from Data with A Numeric-Symbolic Hybrid Deep Network*, arXiv:1812.04426, 2018.

PDE-NET: LEARNING PDES FROM DATA

- As data getting easier and easier to collect, with more and more computing power available, can we learn principles (e.g. PDEs) from data?



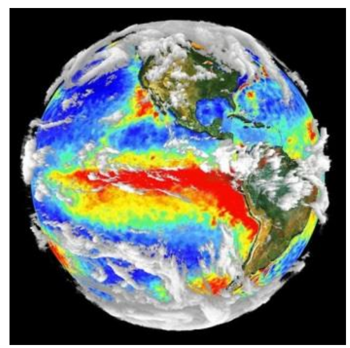
Dynamics of actin in Immunocytoskeleton



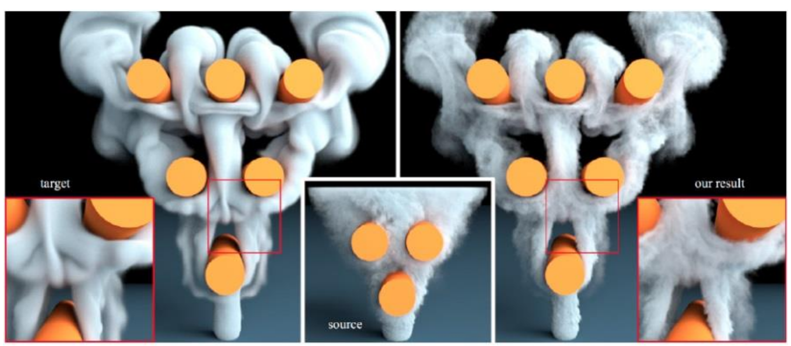
Dynamics of Mitochondria

PDE-Net: Learning PDEs from Data

- As data getting easier and easier to collect, with more and more computing power available, can we learn principles (e.g. PDEs) from data?



Meteorology



Computer Graphics

S. Sato et al., Siggraph 2018

PDE-Net: Learning PDEs from Data

- Earlier work
  - Dictionary based sparse regression
    - Construct dictionary

$$\Theta(U) = [1 \quad U \quad U^2 \quad \cdots U_x \quad UU_x \quad \cdots U_x^2]$$

    - Fit variable ξ

$$U_t = \Theta(U)\xi$$

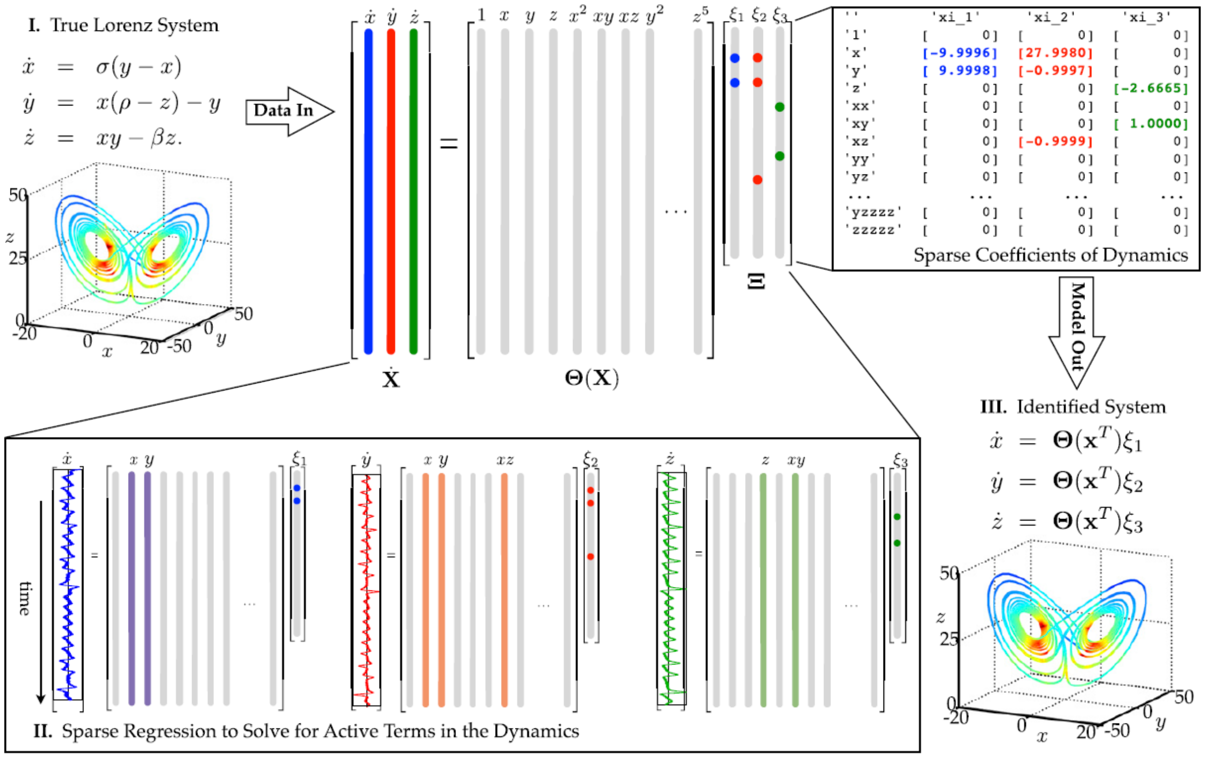
    - Sparse regression

$$\min_{\xi} ||\Theta\xi - U_t||_2^2 + \lambda||\xi||_0$$

- S. Brunton, J. L. Proctor and J. N. Kutz Proceedings of the National Academy of Sciences, 2016
- Samuel H Rudy, Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Science Advances, 3(4), 2017.
- Hayden Schaeffer. Proc. R. Soc. A, volume 473, The Royal Society, 2017.

PDE-NET: LEARNING PDES FROM DATA

- Earlier work
  - Dictionary based sparse regression



- S. Brunton, J. L. Proctor and J. N. Kutz Proceedings of the National Academy of Sciences, 2016
- Samuel H Rudy, Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Science Advances, 3(4), 2017.
- Hayden Schaeffer. Proc. R. Soc. A, volume 473, The Royal Society, 2017.

PDE-Net: Learning PDEs from Data

- Earlier work
  - Coefficients regression
    - Consider

$$u_t + N(u;\lambda) = 0, x \in \Omega, t \in [0,T]$$

      - For example, Burgers equation: $N(u;\lambda) = \lambda_1 u u_x - \lambda_2 u_{xx}$
    - Approximate u by deep neural network(continuous time model)

$$u := Net(t,x), x \in \Omega, \mathrm{t} \in [0,T]$$
$$f := u_t + N(u;\lambda)$$

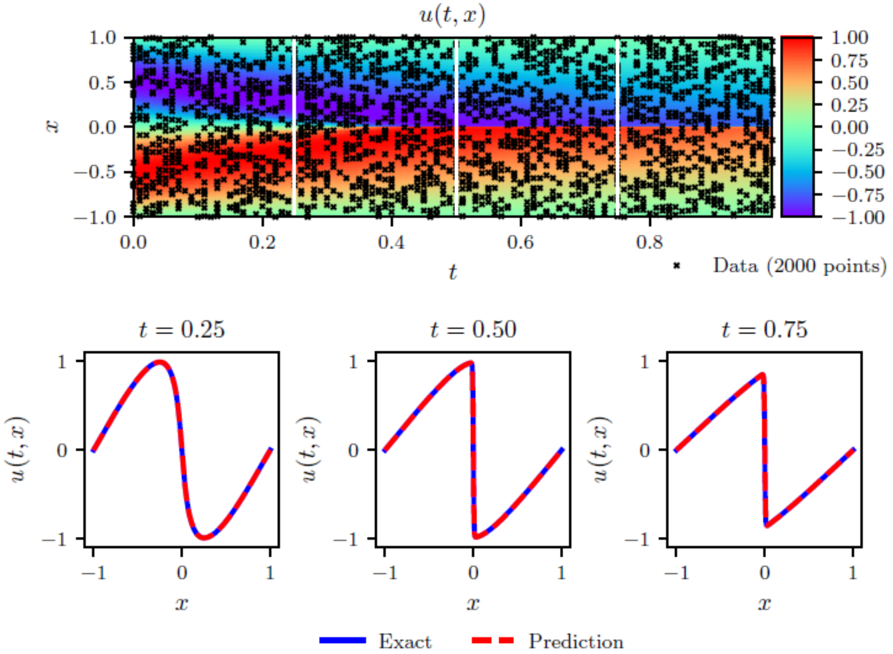
    - Loss function: $MSE = MSE_u + MSE_f$, where

$$MSE_u = \frac{1}{N}\sum_{i=1}^{N}|u(t_u^i, x_u^i) - u^i|^2, \qquad MSE_f = \frac{1}{N}\sum_{i=1}^{N}|f(t_u^i, x_u^i)|^2.$$

- *M. Raissi, P. Perdikaris and G. E. Karniadakis, arXiv preprint arXiv:1711.10566, 2017*

PDE-Net: Learning PDEs from Data

- Earlier work
  - Coefficients regression



Correct PDE	$u_t + uu_x - 0.0031831u_{xx} = 0$
Identified PDE (clean data)	$u_t + 0.99915uu_x - 0.0031794u_{xx} = 0$
Identified PDE (1% noise)	$u_t + 1.00042uu_x - 0.0032098u_{xx} = 0$

- *M. Raissi, P. Perdikaris and G. E. Karniadakis, arXiv preprint arXiv:1703.10230, 2017*

PDE-Net: Learning PDEs from Data

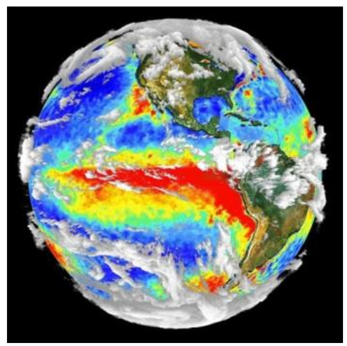
- Remaining challenge
  - Can we go beyond sparse coding framework (linear dictionary)?

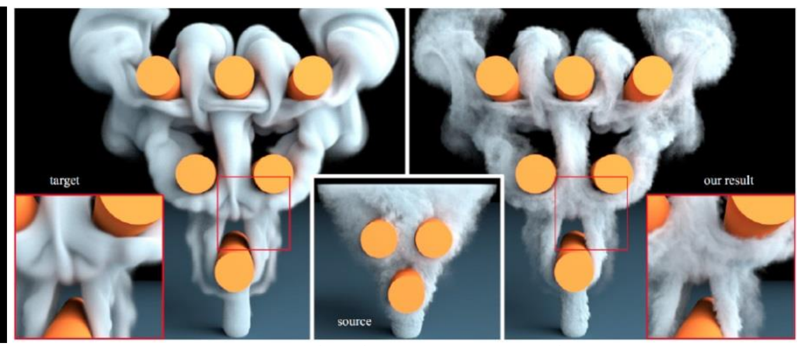
    —— Bigger model class with less prior knowledge
  - Can we learn discrete forms of differential operators and does it help?

    ——More accurate estimation of the PDE and prediction

PDE-Net: Learning PDEs from Data

- As data getting easier and easier to collect, with more and more computing power available, can we learn principles (e.g. PDEs) from data?





S. Sato et al., Siggraph 2018

- Preliminary attempt:
  - Combining deep learning and numerical PDEs
- Objectives:
  - Predictive and expressive power (deep learning)
  - Transparency: to reveal hidden physics (numerical PDEs)

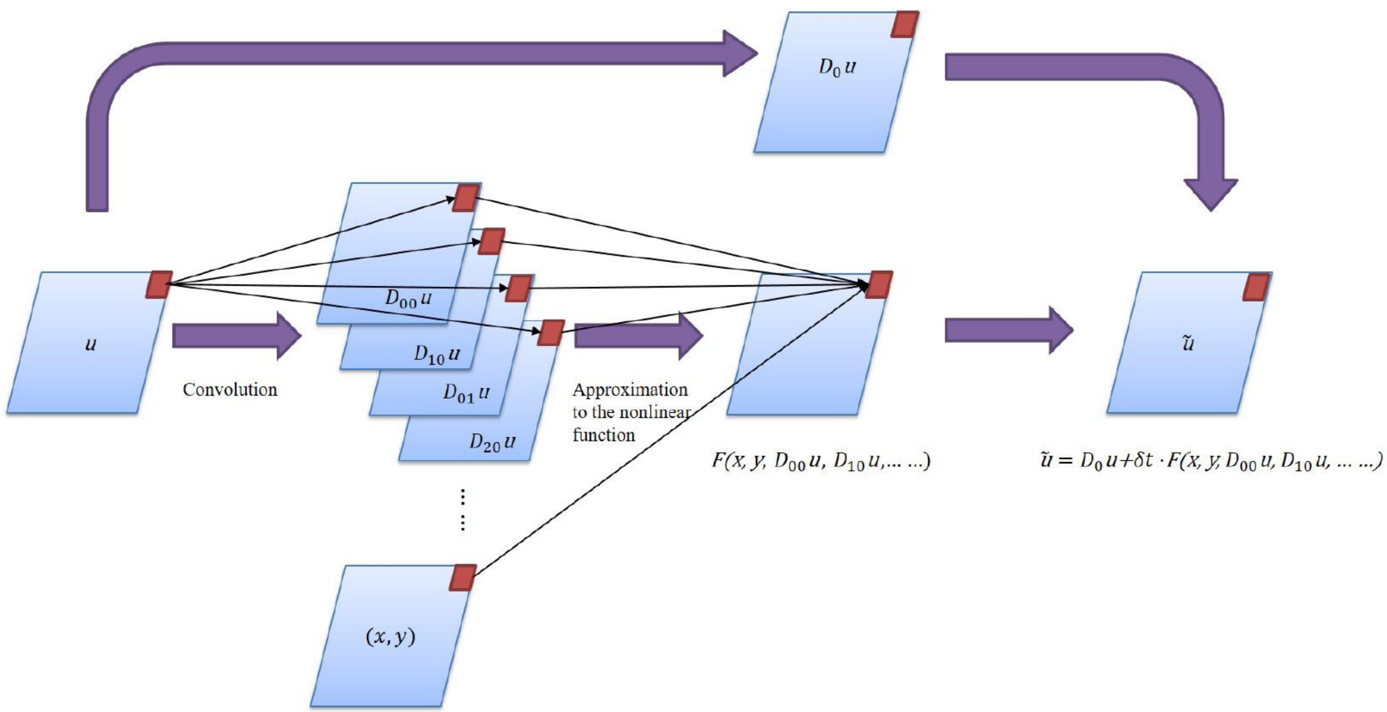
PDE-Net: Learning PDEs from Data

- PDE-Net: a flexible and transparent deep network

Assuming: $$\frac{\partial u}{\partial t} = F(x, u, \nabla u, \nabla^2 u, \dots)$$

Prior knowledge on F:
- Type of the PDE
- Maximum order

- PDE-Net: multi-layer δt-blocks



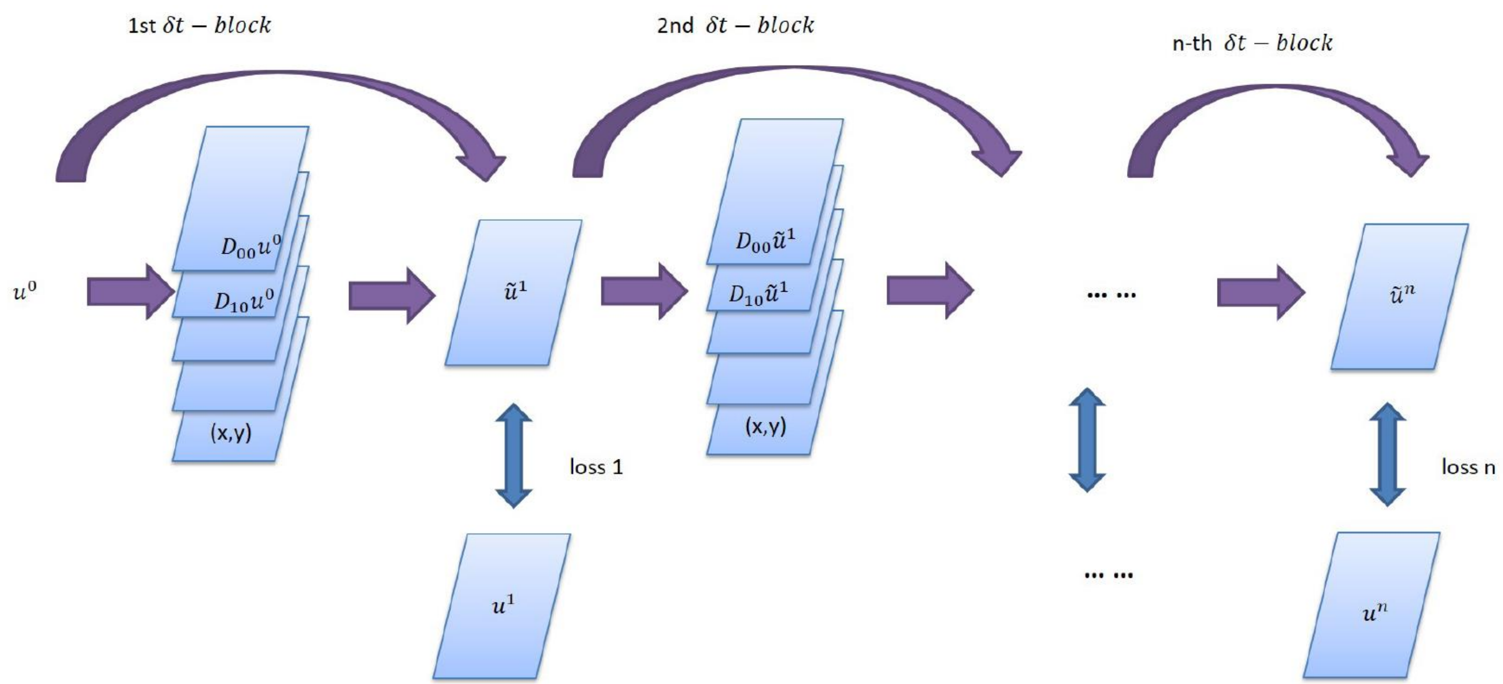
PDE-Net: Learning PDEs from Data

- PDE-Net: a flexible and transparent deep network

Assuming: $$\frac{\partial u}{\partial t} = F(x, u, \nabla u, \nabla^2 u, \dots)$$

Prior knowledge on F:
- Type of the PDE
- Maximum order

- PDE-Net: multi-layer δt-blocks



PDE-Net: Learning PDEs from Data

- Constraints on kernels (granting transparency)
  - Wavelet transform and differential operators

Proposition: *Let q be a filter with sum rules of order $\alpha \in \mathbb{Z}_+^2$. Then for a smooth function $F(x)$ on $\mathbb{R}^2$, we have*

$$\frac{1}{\varepsilon^{|\alpha|}} \sum_{k \in \mathbb{Z}^2} q[k] F(x + \varepsilon k) = C_\alpha \frac{\partial^\alpha}{\partial x^\alpha} F(x) + O(\varepsilon), \text{ as } \varepsilon \to 0.$$

If, in addition, q has total sum rules of order $K \backslash \{|\alpha| + 1\}$ for some $K > |\alpha|$, then

$$\frac{1}{\varepsilon^{|\alpha|}} \sum_{k \in \mathbb{Z}^2} q[k] F(x + \varepsilon k) = C_\alpha \frac{\partial^\alpha}{\partial x^\alpha} F(x) + O(\varepsilon^{K - |\alpha|}), \text{ as } \varepsilon \to 0.$$

$$\begin{aligned}
&\sum_{k_1, k_2 = -\frac{N-1}{2}}^{\frac{N-1}{2}} q[k_1, k_2] f(x + k_1 \delta x, y + k_2 \delta y) \\
= &\sum_{k_1, k_2 = -\frac{N-1}{2}}^{\frac{N-1}{2}} q[k_1, k_2] \sum_{i,j=0}^{N-1} \left.\frac{\partial^{i+j} f}{\partial^i x \partial^j y}\right|_{(x,y)} \frac{k_1^i k_2^j}{i! j!} \delta x^i \delta y^j + o(|\delta x|^{N-1} + |\delta y|^{N-1}) \\
= &\sum_{i,j=0}^{N-1} m_{i,j} \delta x^i \delta y^j \cdot \left.\frac{\partial^{i+j} f}{\partial^i x \partial^j y}\right|_{(x,y)} + o(|\delta x|^{N-1} + |\delta y|^{N-1}).
\end{aligned}$$

- **J.F. Cai, B. Dong, S. Osher and Z. Shen, Journal of the American Mathematical Society, 2012.**
- **B. Dong, Q. Jiang and Z. Shen, Multiscale Modeling & Simulation, 2017**

PDE-Net: Learning PDEs from Data

- Constraints on kernels (granting transparency)
  - Moment matrix

$$M(q) = (m_{i,j})_{N\times N}, \text{ where } m_{i,j} = \frac{1}{(i-1)!(j-1)!}\sum_{k\in\mathbb{Z}^2} k_1^{i-1}k_2^{j-1}q[k_1,k_2]$$

  - We can approximate any differential operator at any prescribed order by constraining $M(q)$
  - For example: approximation of $\frac{\partial f}{\partial x}$ with a 3×3 kernel

$$\begin{pmatrix} 0 & 0 & \star \\ 1 & \star & \star \\ \star & \star & \star \end{pmatrix} \qquad \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & \star \\ 0 & \star & \star \end{pmatrix} \qquad \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

1st order learnable	2st order learnable	1st order frozen

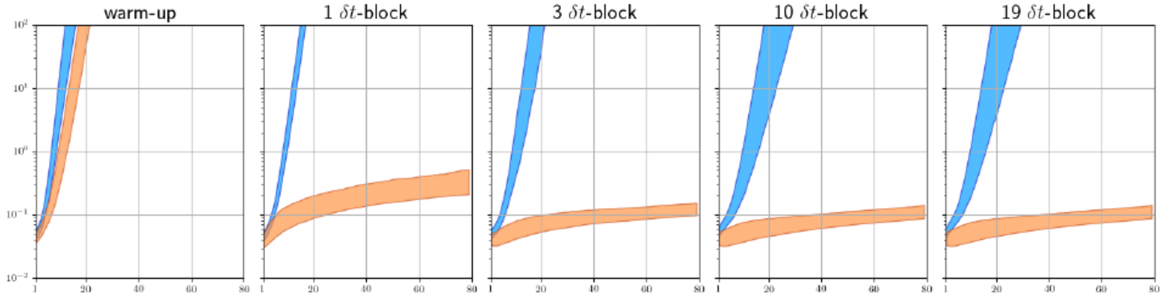
- J.F. Cai, B. Dong, S. Osher and Z. Shen, Journal of the American Mathematical Society, 2012.
- B. Dong, Q. Jiang and Z. Shen, Multiscale Modeling & Simulation, 2017

PDE-NET: LEARNING PDEs FROM DATA

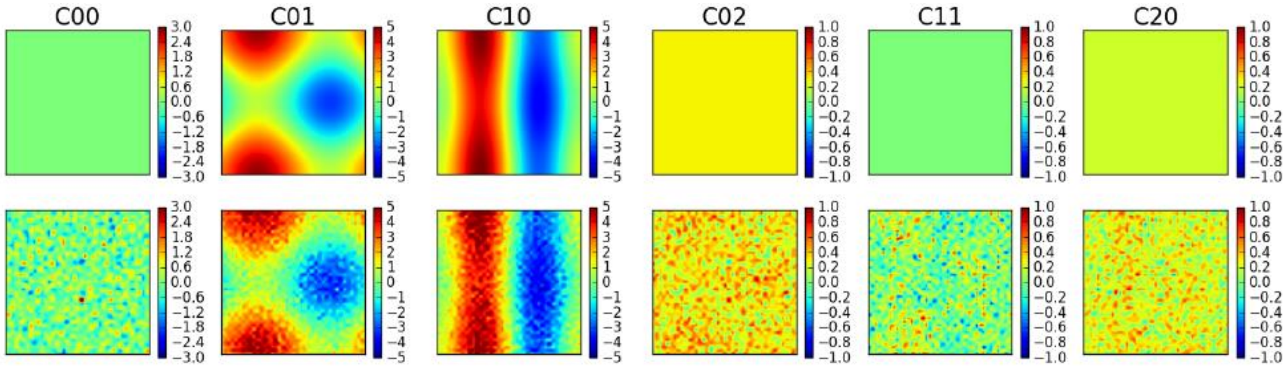
$$\begin{cases} \frac{\partial u}{\partial t} &= a(x,y)u_x + b(x,y)u_y + 0.2u_{xx} + 0.3u_{yy}, \\ u|_{t=0} &= u_0(x,y), \end{cases}$$

$$a(x,y) = 0.5(\cos(y) + x(2\pi - x)\sin(x)) + 0.6,$$
$$b(x,y) = 2(\cos(y) + \sin(x)) + 0.8.$$

- Numerical experiments: results
  - Prediction



  - Model estimation

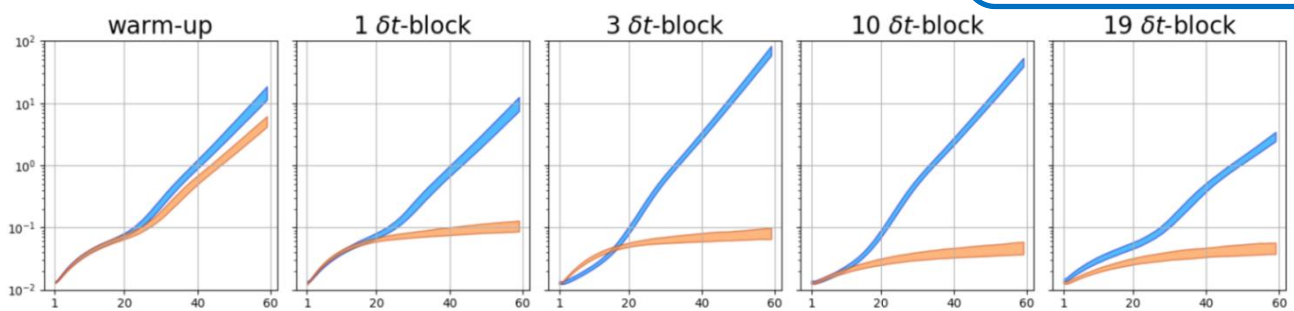


PDE-NET: LEARNING PDES FROM DATA

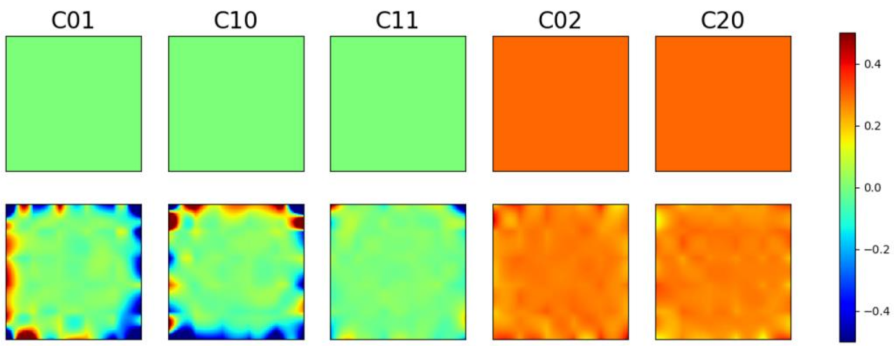
$$\begin{cases} \frac{\partial u}{\partial t} & = c\Delta u + f_s(u) \\ u|_{t=0} & = u_0(x,y), \end{cases}$$

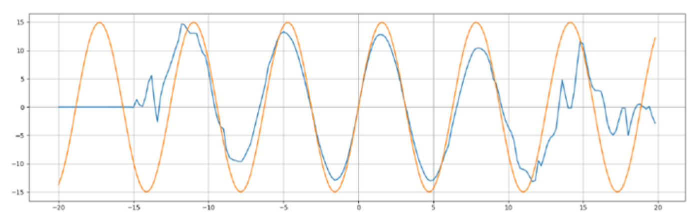
$c = 0.3$ and $f_s(u) = 15\sin(u)$

- Numerical experiments: results
  - Prediction



  - Model estimation





Estimation of the parameters and $f_s(u)$

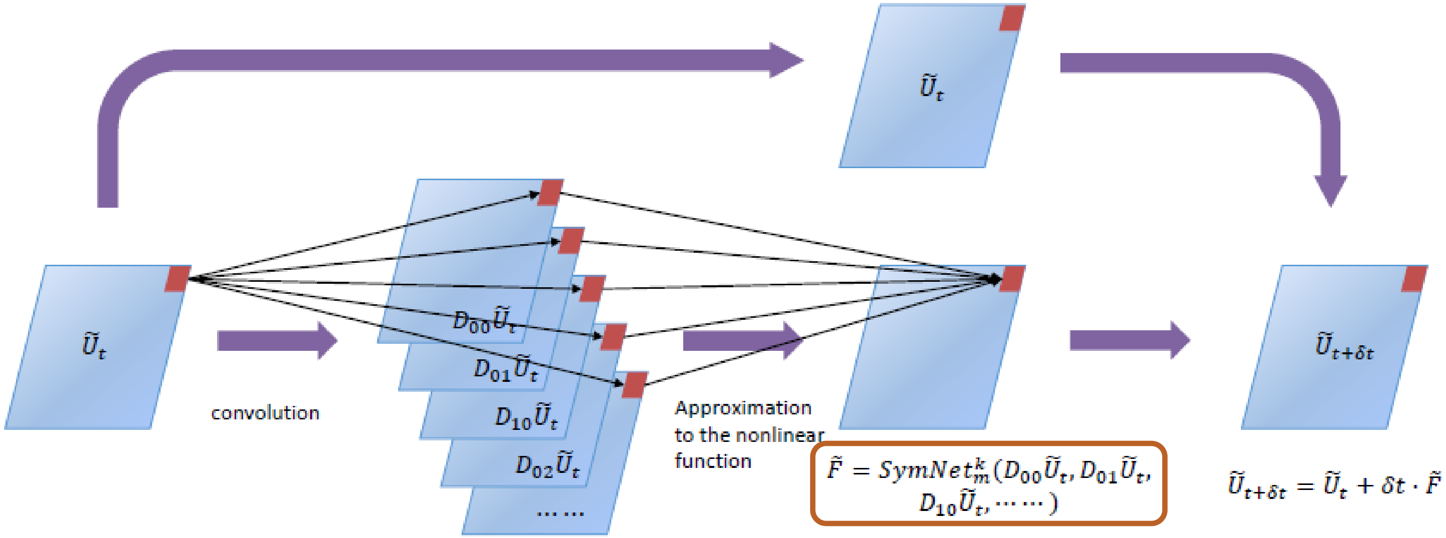
PDE-NET 2.0: COMBINING NUMERIC AND SYMBOLIC REPRESENTATIONS

- Symbolic network (granting transparency)

Assuming: $$\frac{\partial u}{\partial t} = F(u, \nabla u, \nabla^2 u, \dots), \qquad u \in \mathbb{R}^d$$

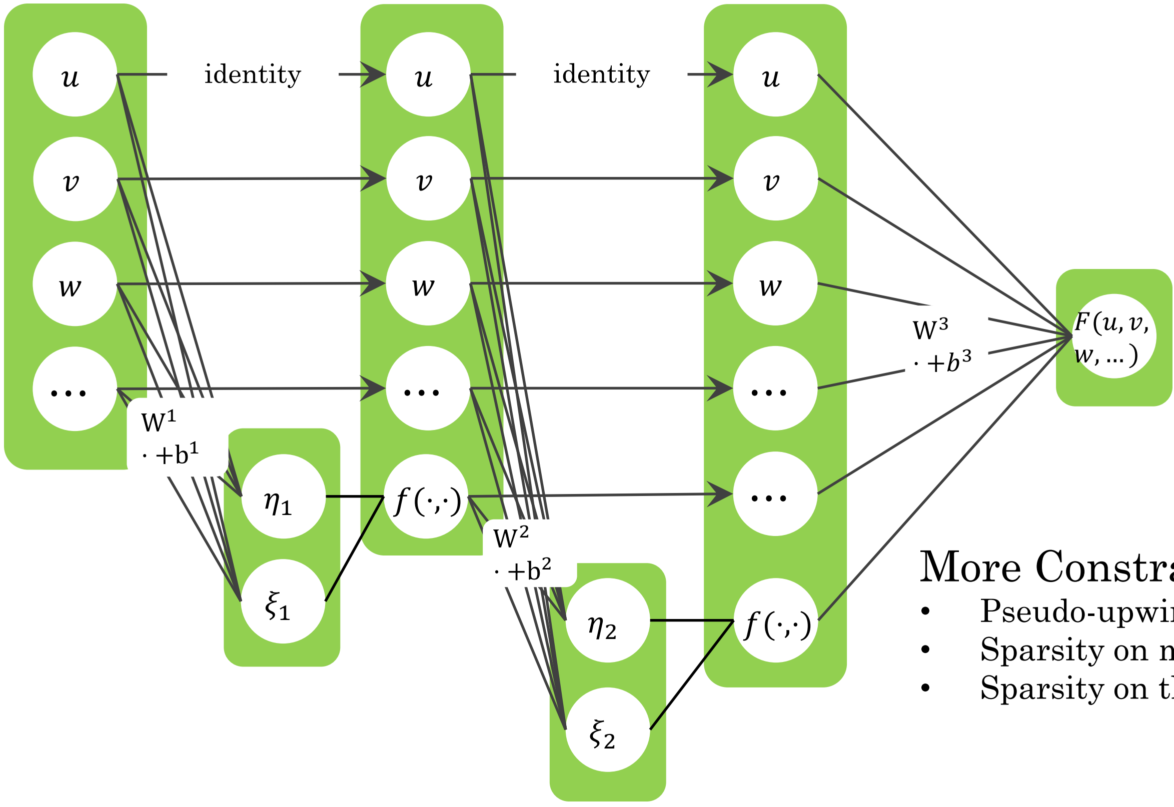
Prior knowledge on F:

- Addition and multiplication of the terms;
- Maximum order.



PDE-NET 2.0: COMBINING NUMERIC AND SYMBOLIC REPRESENTATIONS

- Symbolic network (granting transparency)



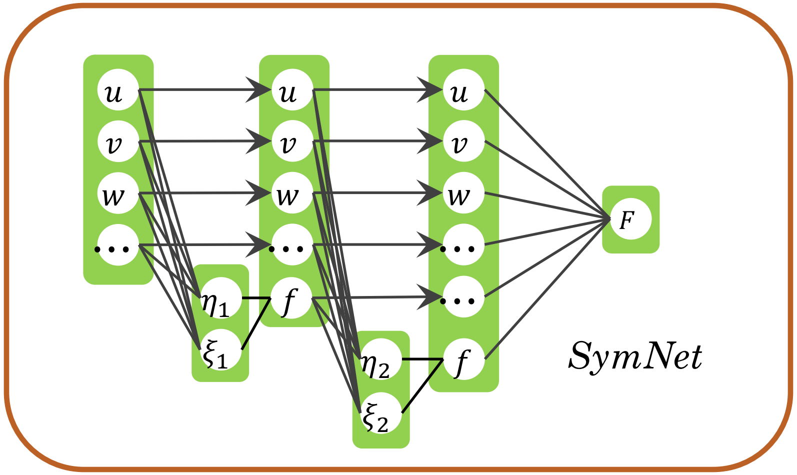
More Constraints:

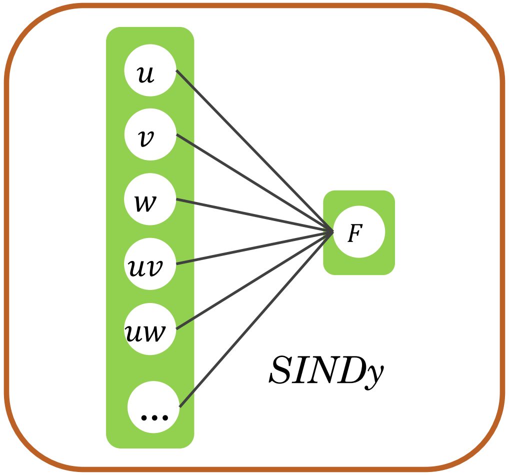
- Pseudo-upwind
- Sparsity on moment matrices
- Sparsity on the symbolic network

Similar to $EQL/EQL^{\div}$: Sahoo, Lampert, and Martius, ICML 2018.

PDE-Net 2.0: Combining Numeric and Symbolic Representations

- Efficiency of the symbolic network
  - **Proposition:** Let $P \in P^k[x_1, \dots, x_m]$ and suppose P have monomials of degree $\leq l$.
    - The memory load of $SymNet_m^k$ that approximates P is $O(m + k)$. The number of flops for evaluating it is $O\big(k(m + k)\big)$.
    - Constructing a dictionary with all possible polynomials of degree l requires $O\left(\binom{m+l}{l}\right)$.





PDE-NET 2.0: COMBINING NUMERIC AND SYMBOLIC REPRESENTATIONS

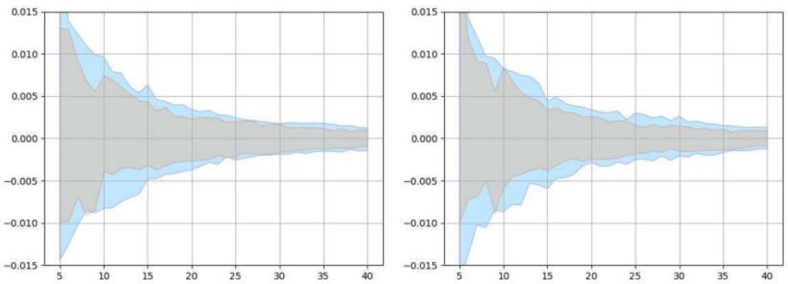
- Fully unknown F
  - Example: Burger's equation

$$\partial_t \boldsymbol{u} + (\boldsymbol{u} \cdot \nabla)\, \boldsymbol{u} = \nu \nabla^2 \boldsymbol{u}$$

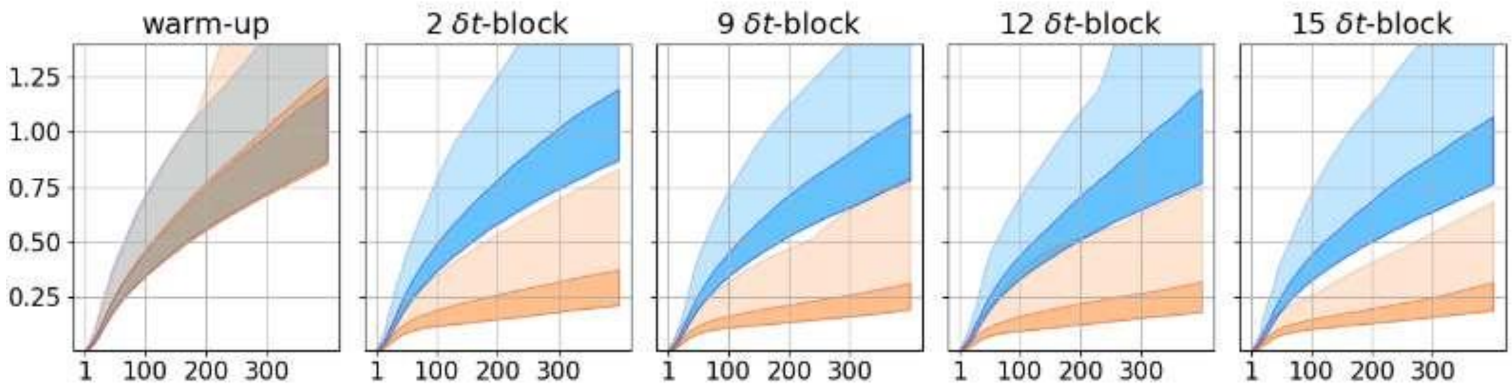
$\nu = 0.05$

Correct PDE	$u_t = -uu_x - vu_y + 0.05(u_{xx} + u_{yy})$ $v_t = -uv_x - vv_y + 0.05(v_{xx} + v_{yy})$
Frozen-PDE-Net 2.0	$u_t = -0.906uu_x - 0.901vu_y + 0.033u_{xx} + 0.037u_{yy}$ $v_t = -0.907vv_y - 0.902uv_x + 0.039v_{xx} + 0.032v_{yy}$
PDE-Net 2.0	$u_t = -0.986uu_x - 0.972u_yv + 0.054u_{xx} + 0.052u_{yy}$ $v_t = -0.984uv_x - 0.982vv_y + 0.055v_{xx} + 0.050v_{yy}$

Model recovery



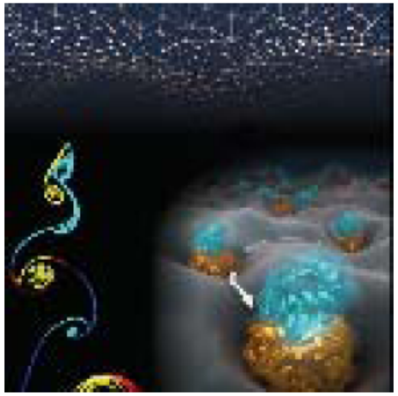
Remainer weights of $\boldsymbol{u}, \boldsymbol{v}$



Prediction

Recent Developments

- IPAM knows better!



Machine Learning for Physics and the Physics of Learning. September 4-December 8, 2019.

- Tutorials: September 5-10, 2019.
- Workshop I: From Passive to Active: Generative and Reinforcement Learning with Physics. September 23-27, 2019.
- Workshop II: Interpretable Learning in Physical Sciences. October 14-18, 2019.
- Workshop III: Validation and Guarantees in Learning Physical Models: from Patterns to Governing Equations to Laws of Nature. October 28-November 1, 2019.
- Workshop IV: Using Physical Insights for Machine Learning. November 18-22, 2019.

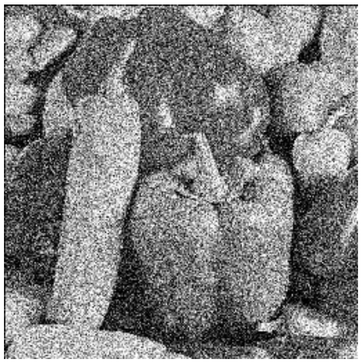
Bridging Differential Equations with Deep Networks

An application in image restoration

- Xiaoshuai Zhang, Yiping Lu, Jiaying Liu and Bin Dong, *Dynamically Unfolding Recurrent Restorer: A Moving Endpoint Control Method for Image Restoration*, ICLR 2019. (arXiv:1805.07709)

Motivation: Denoising with Unknown Noise Level

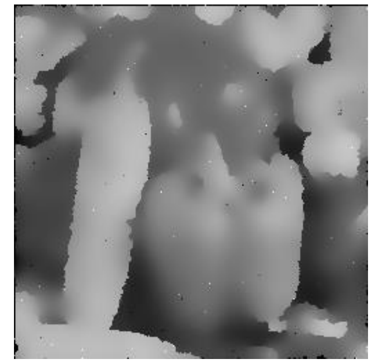
input

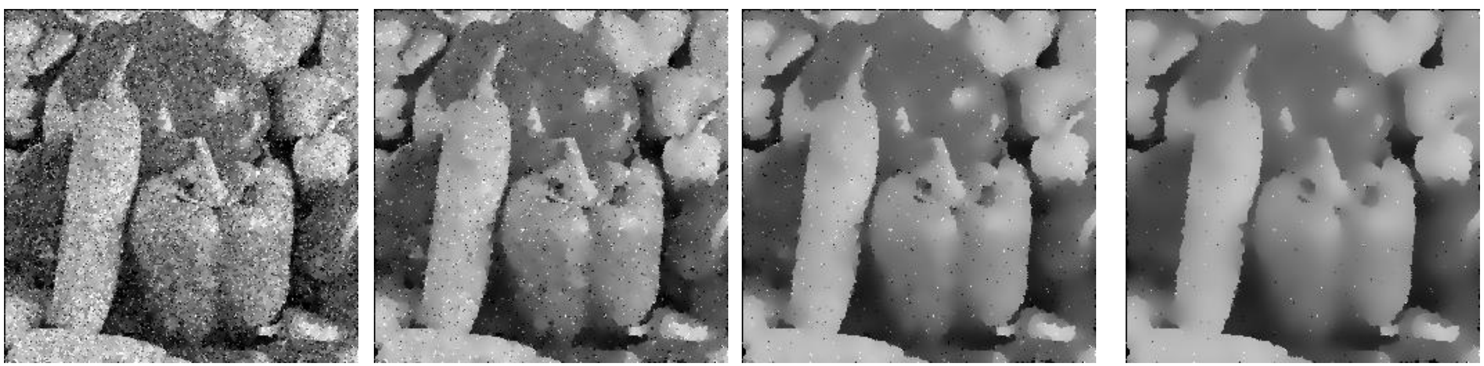


$$\begin{cases} \dfrac{\partial u}{\partial t} = \operatorname{div}\left(c(|\nabla u|^2)\ \nabla u\right) \ \text{ in } \ \Omega \times (0, \mathrm{T}), \\ \dfrac{\partial u}{\partial N} = 0 \ \text{ on } \ \partial\Omega \times (0, T), \\ u(0, x) = u_0(x) \ \text{ in } \ \Omega, \end{cases}$$

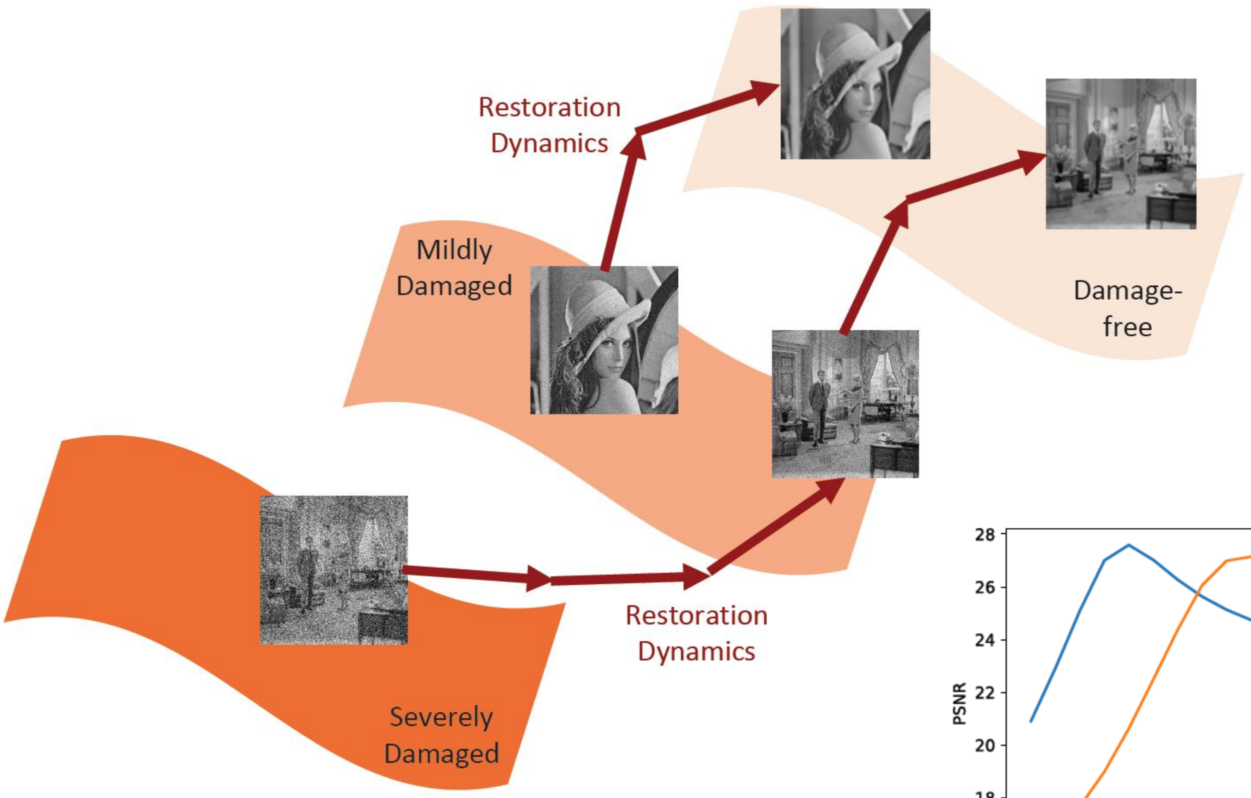
processing

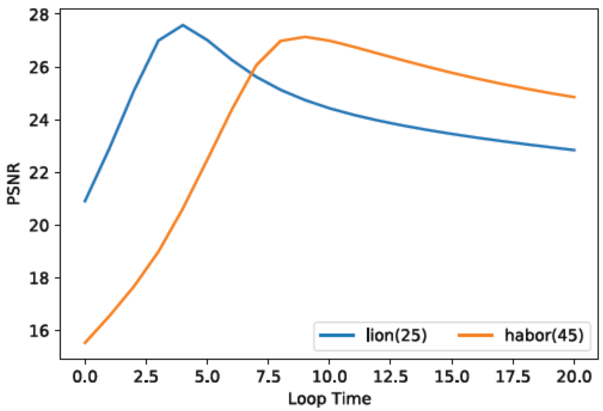
output





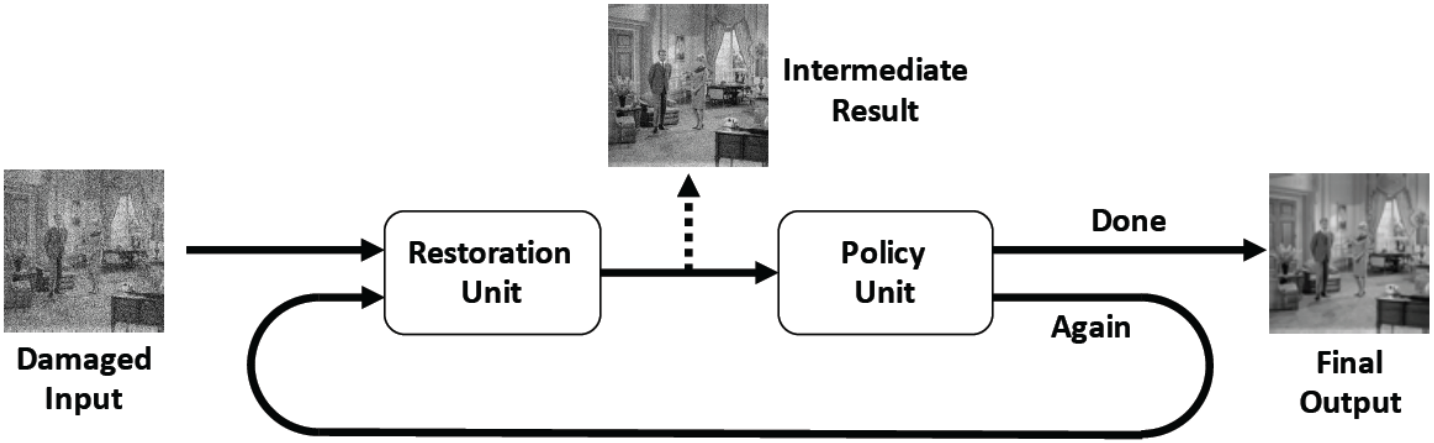
Motivation: Denoising with Unknown Noise Level





Moving Endpoint Control Approach

$$\min_{w,\tau} L(X(\tau), y) + \int_0^\tau R(w(t), t)dt$$
$$s.t.\ \dot{X} = f(X(t), w(t)), t \in (0, \tau)$$
$$X(0) = x_0.$$



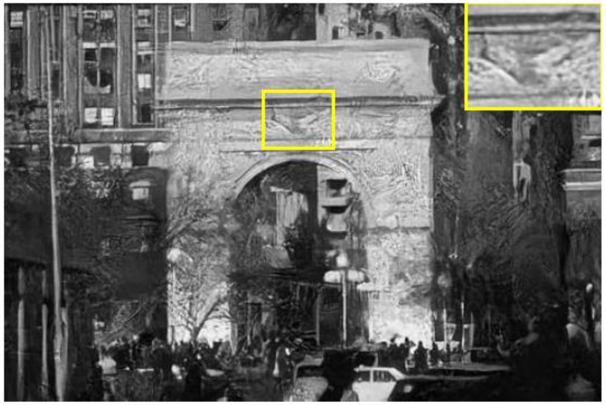
Dynamically Unfolding Recurrent Restorer (**DURR**)

Denoising with Unknown σ

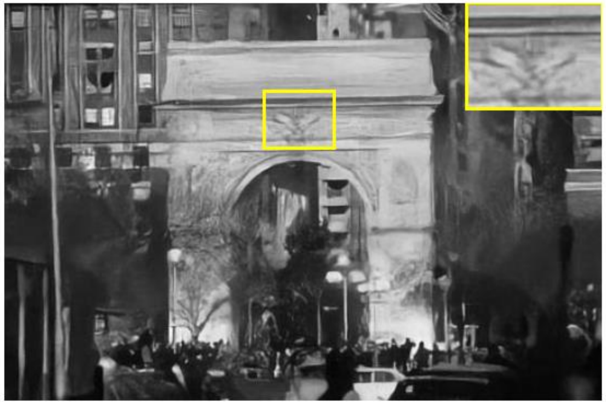
BSD68 Image Denoising Data Set

	BM3D	WNNM	DnCNN-B	UNLNet_5	DURR
$\sigma = 25$	28.55	28.73	29.16	28.96	29.16
$\sigma = 35$	27.07	27.28	27.66	27.50	27.72
$\sigma = 45$	25.99	26.26	26.62	26.48	26.71
$\sigma = 55$	25.26	25.49	25.80	25.64	25.91
$\sigma = 65$	24.69	24.51	23.40*	-	25.26*
$\sigma = 75$	22.63	22.71	18.73*	-	24.71*

Generalization beyond training noise level!



DnCNN, 21.86dB

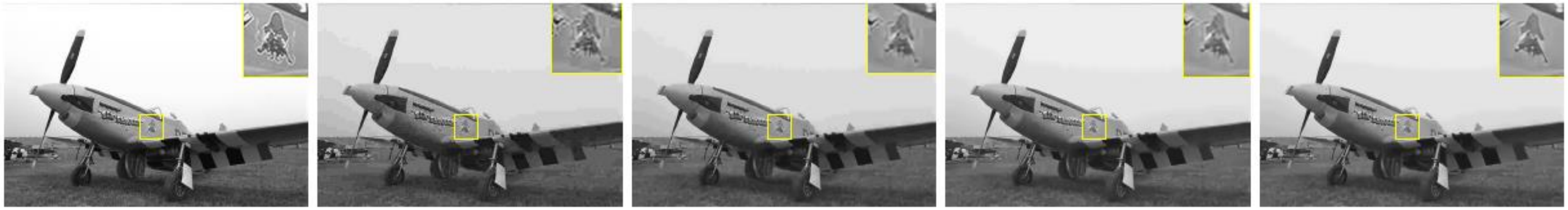


DURR, 22.84dB

DnCNN-B: K. Zhang et al. TIP 2017; UNLNet: S Lefkimmiatis, CVPR 2018

JPEG Deblocking with Unknown QFs

QF	JPEG	SA-DCT	AR-CNN	AR-CNN-B	DnCNN-3	DURR
10	27.77	28.65	28.98	28.53	29.40	29.23*
20	30.07	30.81	31.29	30.88	31.59	31.68
30	31.41	32.08	32.69	32.31	32.98	33.05
40	32.45	32.99	33.63	33.39	33.96	34.01*



Ground Truth JPEG (a) AR-CNN (b) DnCNN (c) DURR

SA-DCT: A. Foi et al., TIP 2007;
AR-CNN: C. Dong et al., CVPR 2015;
DnCNN-3: K. Zhang et al., TIP 2017.

Some Related Works

- Jin M, Roth S, Favaro P. *Noise-blind image deblurring*. CVPR 2017.

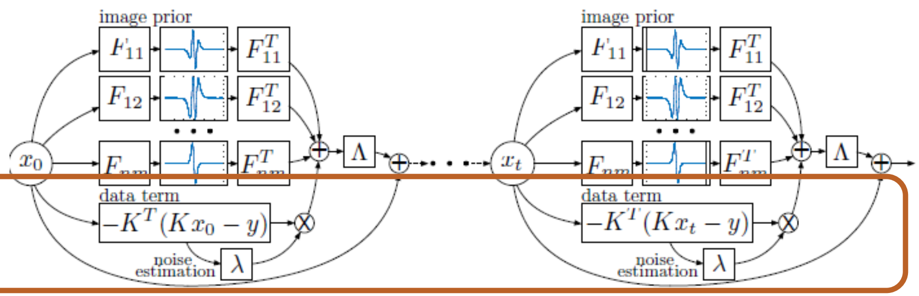
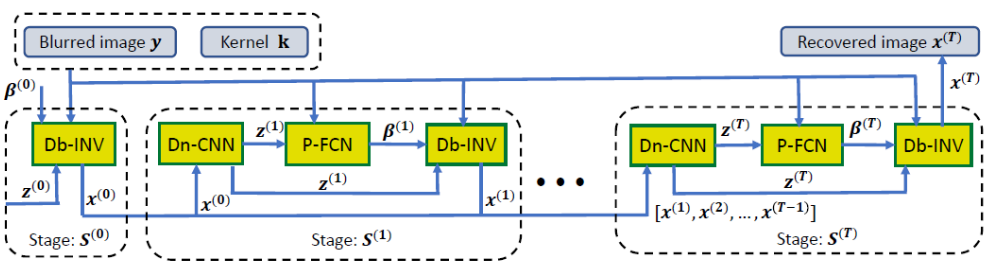
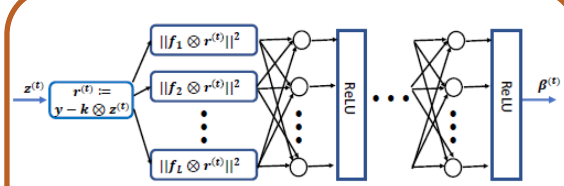


Figure 2. The GradNet architecture.

- Y. Nan, Y. Quan, H. Ji, Learning for Non-blind Deconvolution: The Devil Is STILL in Details, preprint 2019.





(a) Diagram of P-FCN for defining $\{\beta_i^{(t)}\}$

Conclusions

- We suggested a (heuristic) bridge between numerical differential equations and deep neural architectures, and proposed some new architectures for different tasks.
- Future directions:
  - Theoretical analysis of architectures: optimization, generalization, recovery guarantees, etc.
  - Robustness and compactness of DNNs.
  - Learning "principles" from data (beyond PoC).

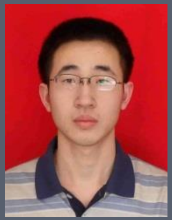
Jointly with

Quanzheng Li
@ Harvard MGH

Jiaying Liu
@ PKU ICST

Zichao Long
@ PKU SMS

Yiping Lu $\binom{x}{2}$
@ PKU SMS



Xianzhong Ma
@ PKU SMS

Xiaoshuai Zhang
@ PKU ICST

Aoxiao Zhong
@ Harvard SEAS

Thanks for Your Attention!

My Webpage:
http://bicmr.pku.edu.cn/~dongbin