Deep Learning Surrogate Models for Kinetic Landau Fluid Closure

X. Q. Xu[1],

C. H. Ma[2], L.B. Wang[3], B. Zhu[1]

1) Lawrence Livermore National Laboratory, Livermore, CA 94550 USA
2) Princeton Plasma Physics Laboratory, Princeton, NJ 08543-0451, USA
3) Peking University, Beijing, China

Acknowledgement: T. Bremer, L. Peterson, B. Van Essen, X. Z. Tang, and Z. H. Wang

Presented at
NERSC 2020 NUG Annual Meeting
August 17, 2020

This work was performed under the auspices of the U.S. DoE by LLNL under Contract DE-AC52-07NA27344 , LLNL-PRES-xxxxxx

Fluid simulations with kinetic closures can be achieved by Deep Learning, with DL surrogate model closure predicted by the neural networks

Kinetic model:

$$\frac{\partial f}{\partial t} + v\frac{\partial f}{\partial x} + \frac{e}{m}E\frac{\partial f}{\partial v} = 0. \quad (1)$$

Fluid model ($n = \int f dv, mnu = m\int vf dv, p = m\int (v-u)^2 f dv$):

$$\frac{\partial n}{\partial t} + \frac{\partial (un)}{\partial x} = 0, \quad (2)$$

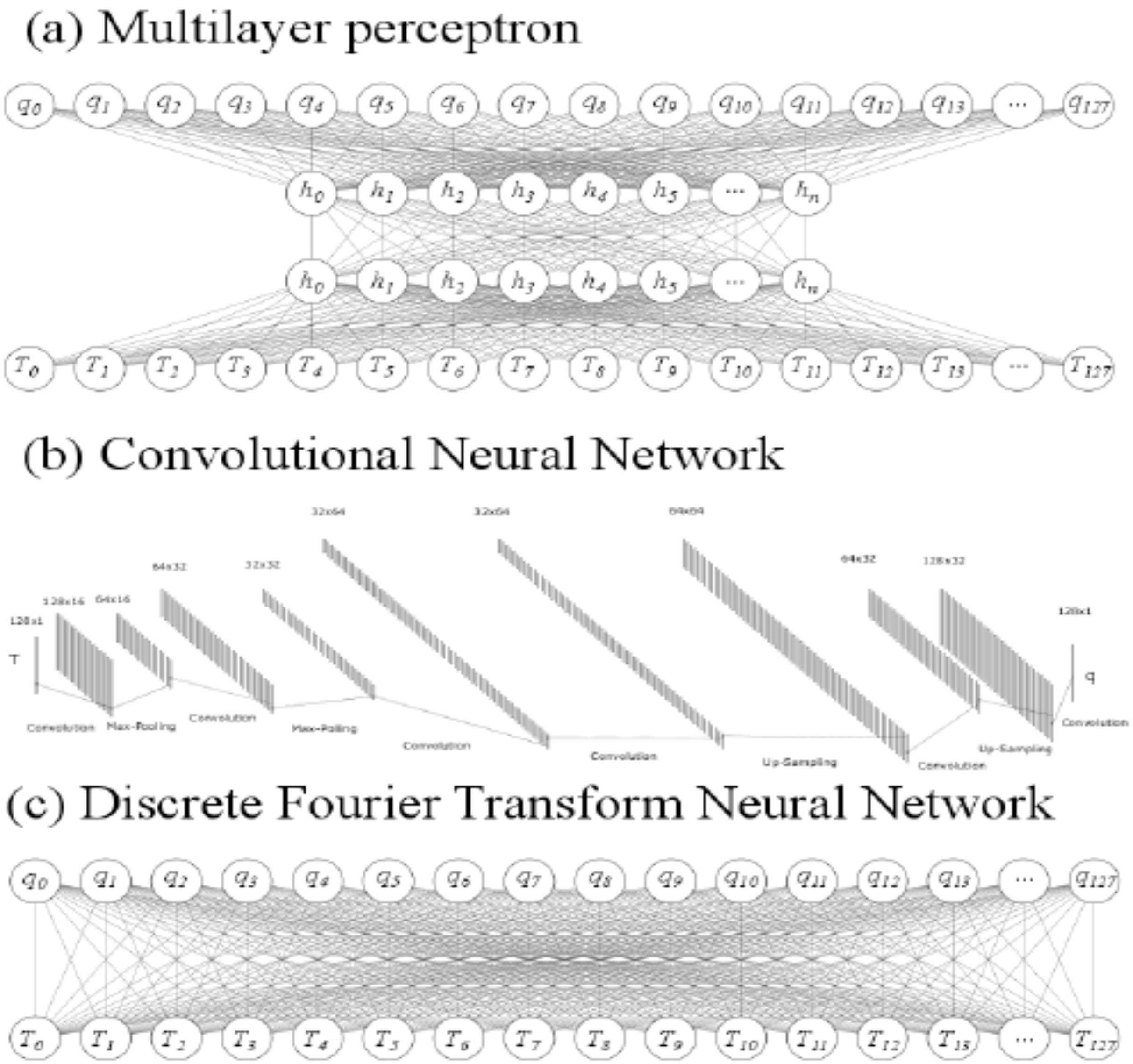
$$\frac{\partial mnu}{\partial t} + \frac{\partial (umnu)}{\partial x} = -\frac{\partial p}{\partial x} + enE, \quad (3)$$

$$\frac{\partial p}{\partial t} + \frac{\partial (up)}{\partial x} = -2p\frac{\partial u}{\partial x} - \frac{\partial q}{\partial x}. \quad (4)$$

Kinetic closure

$$q = \sqrt{2} n_0 v_t \frac{(2\zeta^3 - 3\zeta)\, Z(\zeta) + 2\zeta^2 - 2}{(2\zeta^2 - 1)\, Z(\zeta) + 2\zeta} \hat{k} T \qquad \zeta = \frac{\omega + i\nu}{\sqrt{2}|k| v_t} \quad (5)$$

With such a complicated spatiotemporal closure, how can one implement it into a fluid simulation in configuration space?



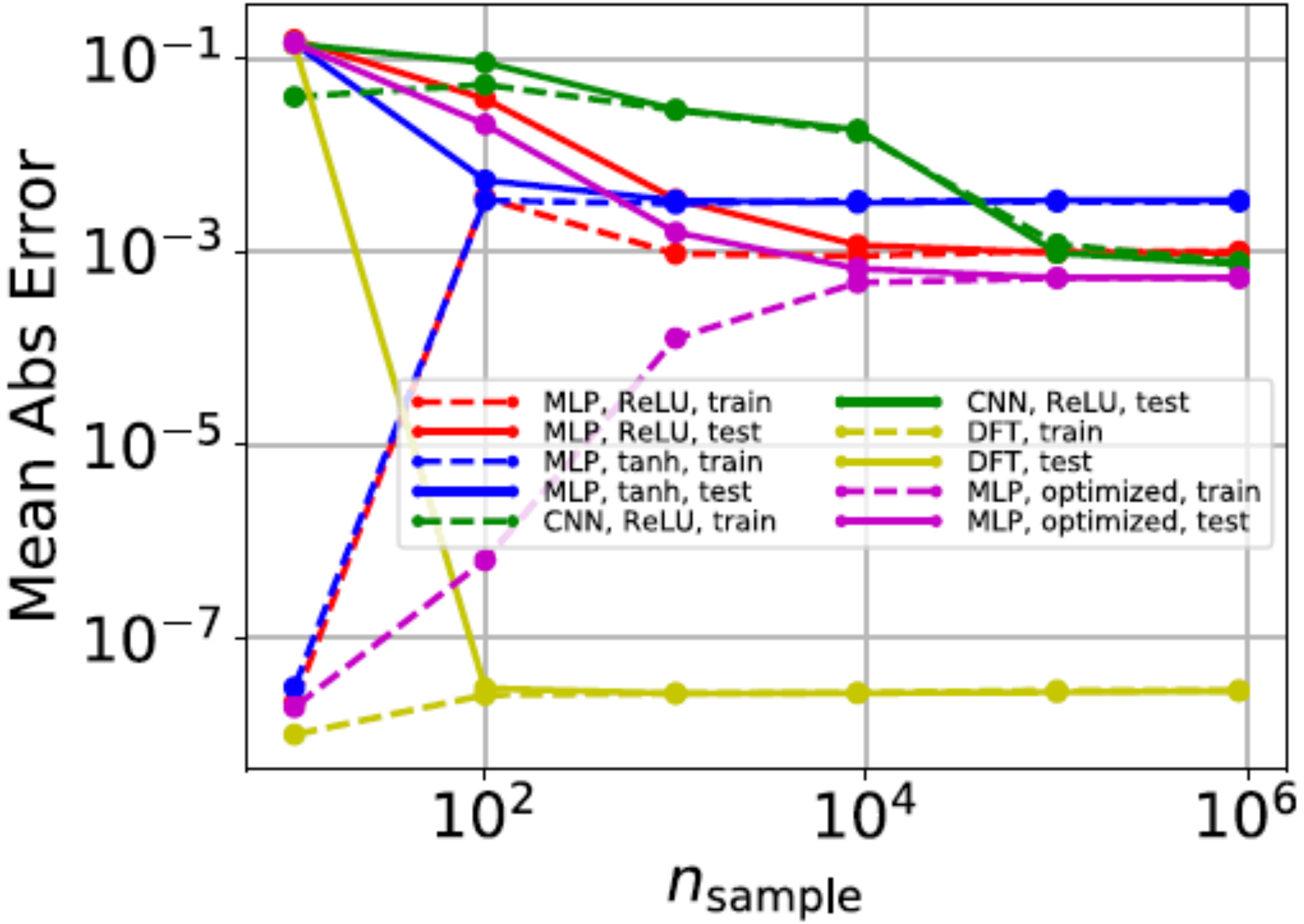
Neural networks:

- MLP: two hidden layers, fully connected
- CNN: Deep partially connected convolutional layers
- DFT: No hidden layers, fully connected

hyperparameters:

- Constructed with Tensorflow and Keras
- Optimizer: Adam
- Loss function: mean-squared-error(MSE)

Mean-Absolute-Error vs the number of training samples for different types of neural networks with different activation functions

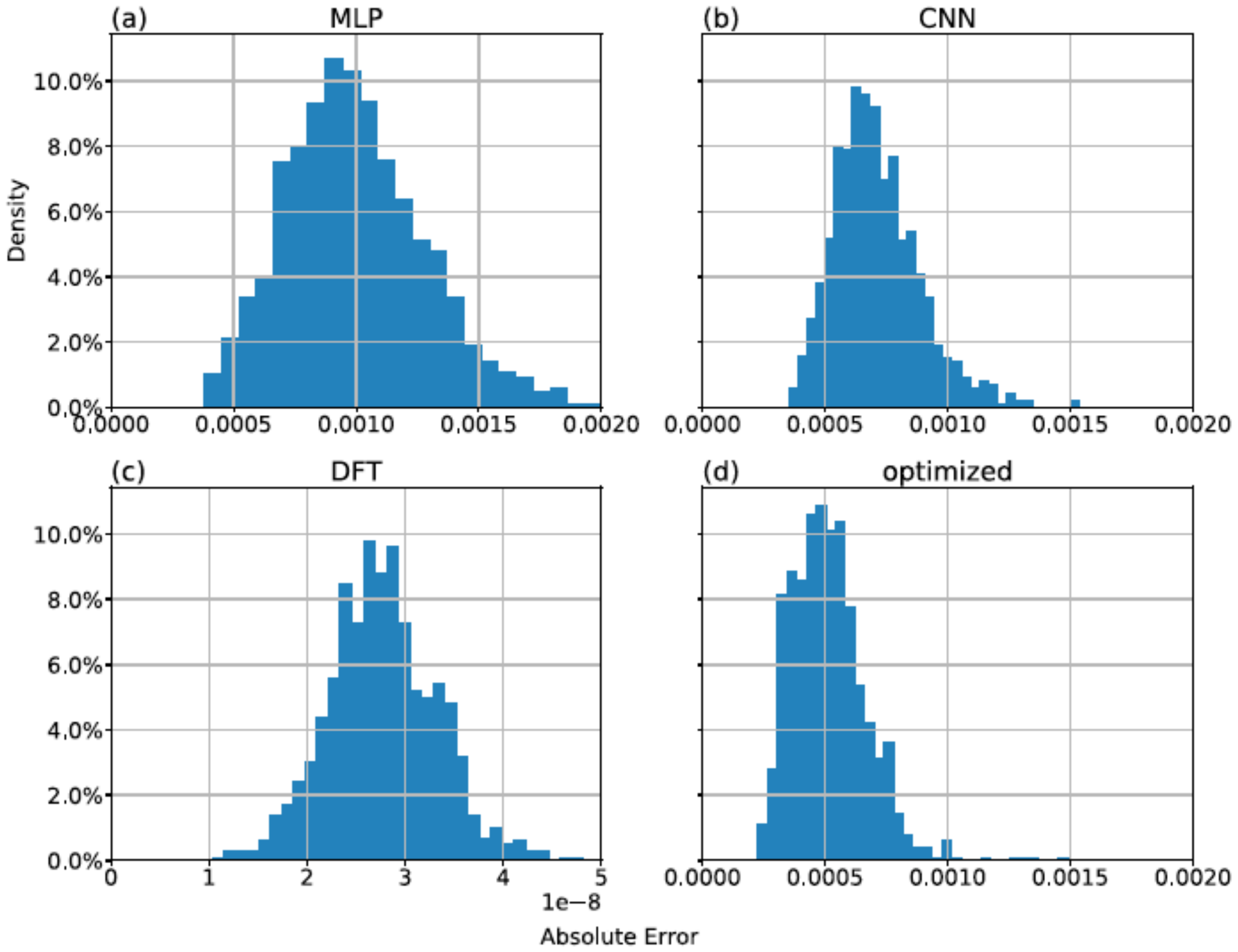


HP Landau closure
Hammett and Perkins PRL 1990

$$q_k = -n_0\sqrt{\frac{8}{\pi}}v_t\frac{ikT_k}{|k|}$$

$$q(x) = -n_0\sqrt{\frac{8}{\pi^3}}v_t\int_0^\infty dx'\left[T(x+x') - T(x-x')\right]/x'.$$

Probability distribution function (PDF) of absolute error for different neural network models when $n_{sample} = 10^6$



The Mean absolute errors for different NN

$\varepsilon_{MLP} = 9.72 \times 10^{-4}$

$\varepsilon_{CNN} = 7.45 \times 10^{-4}$

$\varepsilon_{optimized} = 5.49 \times 10^{-4}$ for MLP

$\varepsilon_{DFT} = 2.69 \times 10^{-8}$

Relative error (MRE) comparison of HP closure between DL surrogate model & non-Fourier (NF)

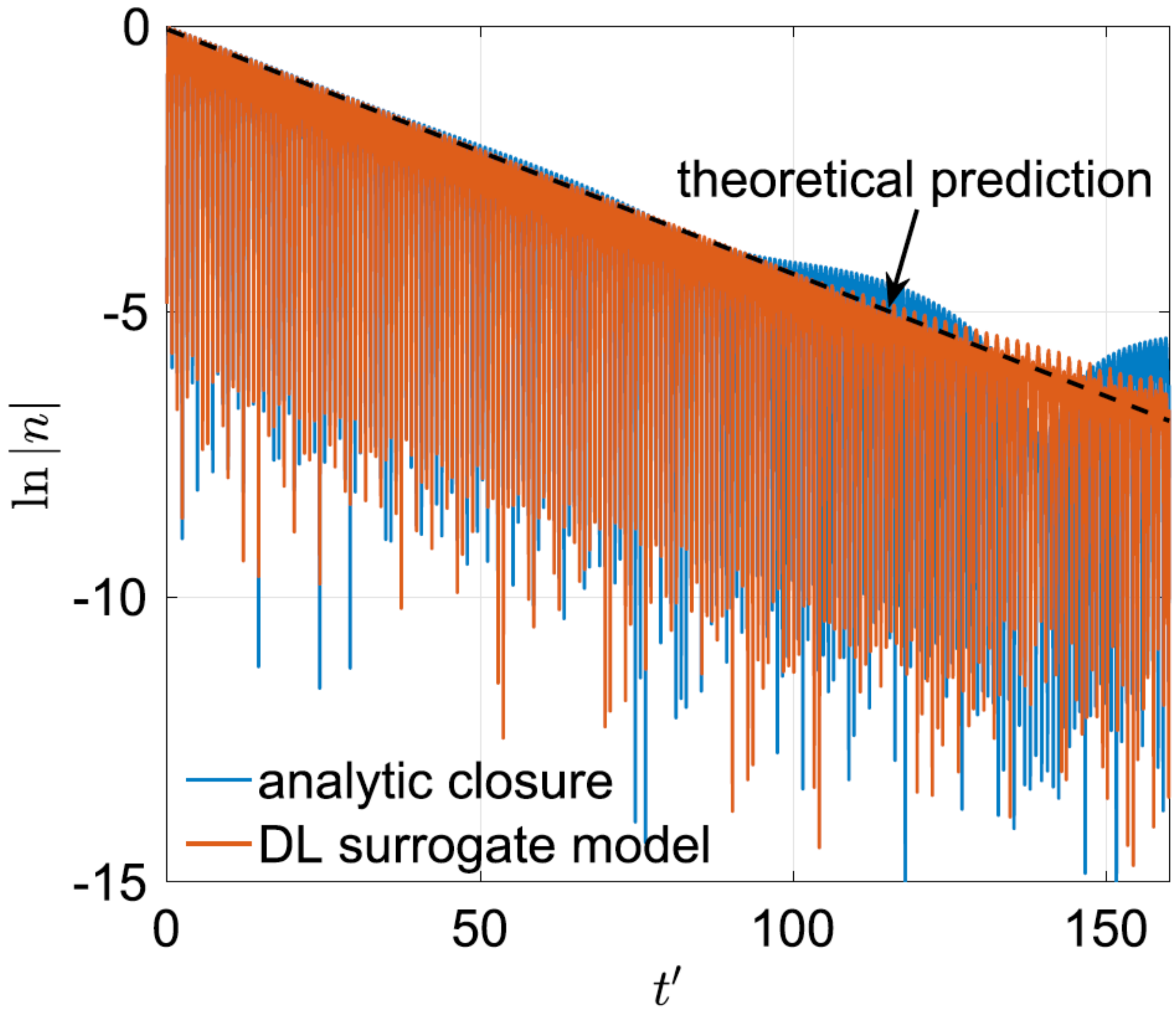
$$\mathrm{MRE} = \frac{1}{n_s N_z} \sum_{i=0}^{n_s-1} \frac{\sum_{j=0}^{N_z-1} |q_{\{\}}[i;j] - q_{kinetic}[i;j]|}{\max(q_{kinetic}[i;:])}$$

N_s=10000, N_z=128, N_{layers}=3, $N_{neuron}=2^{10}$, N_k=7

$k_{\max}$	ν	N_s	MRE for DL	MRE for NF	MRE for NF estimates
4	(0,100)	10^6	0.038%	0.12%	$<$ 0.17%
4	(0,10)	10^6	0.012%	0.12%	$<$ 0.17%
8	(0,100)	10^6	0.062%	0.1%	$<$ 0.15%
8	(0,10)	10^6	0.025%	0.1%	$<$ 0.15%
16	(0,100)	10^6	0.07%	0.13%	$<$ 0.22%
16	(0,10)	10^6	0.025%	0.13%	$<$ 0.22%
8	(0,100)	10^7	0.025%	0.1%	$<$ 0.15%
8	(0,100)	10^7	0.02%(5L)	0.1%	$<$ 0.15%
8	(0,10)	10^7	0.015%	0.1%	$<$ 0.15%
8	(-1,100)	10^7	0.023%	0.1%	$<$ 0.15%
8	(-1,10)	10^7	0.013%	0.1%	$<$ 0.15%

Comparison of fluid simulations between analytical and DL surrogate model closures for collisionless kinetic Langmuir waves

N_s=10000, N_z=128, N_{layers}=3, N_{neuron}=2^{10}, N_k=7



- **For the DL model, there is a concern about the error accumulation problem**
- **Our results show that simulations with the deep learning surrogate model are as good as, if not better than, simulations with the analytic closure in terms of long-term numerical stability in the linear Landau damping test.**

Summary & Discussion

- Appropriate neural networks are capable to calculate Landau fluid closure;

- NN can be implemented in global fluid legacy codes, such as BOUT++ for the LF closure in configuration space

- Training neural network closure with kinetic simulation data:

  Kinetic codes —train→ NN —closure→ BOUT++

- Training more neural network closures for macroscopic fluid simulation codes to capture microdetails, such as
  - turbulence fluxes,
  - Trapped particles
  - …