Deep Learning-based Fokker-Planck-Landau Collision Operator for Fusion Plasma Simulation: A Preliminary Study

HyungJun Noh[a], Jimin Lee[b*], Eisung Yoon[b]
[a]*Department of Physics, Ulsan National Institute of Science and Technology, 50, UNIST-gil, Ulsan, 44919*
[b]*Department of Nuclear Engineering, Ulsan National Institute of Science and Technology, 50, UNIST-gil, Ulsan, 44919*
[*]*Corresponding author: jiminlee@unist.ac.kr*

1. Introduction

Kinetic simulations for nuclear fusion devices such as tokamaks need nonlinear Coulomb collision operators for calculating collisions among multiple plasma species. Considering tungsten wall in ITER, different charge states of tungsten ions that emerged in plasmas would demand to handle many plasma species [1]. However, the collision operators would take a huge computational cost as floating-point operations of the operator grow with $\sim O\left({N_{sp}}^2\right)$, where N_{sp} is the number of species.

In this work, we investigate the feasibility of deep learning (DL)-based FPL collision operators to accelerate collision calculation in the kinetic simulations. Currently in particular, XGC (X-point Gyrokinetic Code) uses a nonlinear Fokker-Planck-Landau (FPL) collision operator based on Finite Volume Method (FVM) [2]. Our preliminary results showed the promising potential that DL model can learn advection and diffusion features of collision operators and ultimately can be a surrogate of physical collision models.

2. Methods and Results

2.1 Fokker-Planck-Landau equation

The FPL equation is written as [2]

$$\frac{df_a}{dt} = \sum_b C_{ab}(f_a; f_b') = -\sum_b \frac{e_a^2 e_b^2 \ln\Lambda_{ab}}{8\pi\epsilon_0^2 m_a} \nabla_v \cdot \int \boldsymbol{U} \cdot \left(\frac{f_a}{m_b}\nabla_v' f_b' - \frac{f_b'}{m_a}\nabla_v f_a\right) d^3v', \quad (1)$$

where C_{ab} denotes the collision operator between species "a" in v coordinate and field species "b" in v' coordinate. e, m, and f denote charge, mass, and probability density function (pdf) of the species respectively, where the species type is represented as subscript a and b. And $\ln\Lambda$ is the Coulomb Logarithm, t is the time and $\boldsymbol{U}$ is a tensor defined with relative velocity $\boldsymbol{u} = v - v'$,

$$\boldsymbol{U} = \frac{u^2\boldsymbol{I} - \boldsymbol{uu}}{u^3}.$$

2.2 Preprocessing the Training Data

Using the FPL solver based on FVM written in MATLAB [2], our raw data on a two-dimensional velocity grid ($N_{\mathrm{v}_\perp} \times N_{\mathrm{v}_\parallel} = 40 \times 60$) was prepared. To train our model to handle various anisotropic temperature cases, we simulated over a broad range of $\mathrm{v}_\parallel$axis temperature ($T_{\mathrm{v}\parallel}$=71eV~99eV) while temperature along $\mathrm{v}_\perp$ axis was fixed to $T_{\mathrm{v}_\perp}$ =100eV. For each condition, simulation was held for consecutive 49 time-steps. Particle's pdf at each time step on the velocity grid was saved and used as training data. A set of 1,296 data were divided to a training set (n=1,008), a validation set (n=144) and a test set (n=144). All input data were normalized to [0,1], and the range of output data was recovered to the original range.

2.3 Model Architecture

Our model uses convolutional neural network (CNN) based encoder-decoder networks [3]. CNN fits our task as it preserves the spatial structure of the data. The overall schematic of model architecture is shown in Fig. 1. The model consists of an encoder that learns the overall context of data while image size is reduced via max-pooling and a decoder that constructs images with localized context. Skip connection concatenates feature maps in the encoder path and upsampled features in the decoder path. This helps our model make use of fine details learned in the encoder path when it constructs an image in the decoder path. For a stable training process, Kaiming He initialization and Batch normalization were applied to each layer with ReLU activation functions, except the last convolutional layer. If a pdf at an arbitrary time step with some anisotropic temperature condition is given, the trained DL-based FPL model produces a pdf at the next time step. 95,585 trainable parameters are used in the neural network.

When training the neural network, model parameters were updated to minimize mean squared error (MSE) loss functions. Our MSE represents the error between FVM and the model's output for a given input and is defined as

$$MSE = \frac{1}{N_{\mathrm{v}_\perp} N_{\mathrm{v}_\parallel}} \sum_{j=1}^{N_{\mathrm{v}_\perp}} \sum_{i=1}^{N_{\mathrm{v}_\parallel}} \left(f_{\mathrm{label}_{i,j}} - f_{\mathrm{DL}_{i,j}}\right)^2, \quad (2)$$

where f_{label} denotes pdf from FVM, namely our training data and f_{DL} is our model's output. i and j indices are used as the pdf is two-dimensional (Sec. 2.2). Adam optimizer was used and learning rate decay optimally decreased learning rate exponentially as epoch

progresses for fast and steady training. To avoid overfitting in which the model is too optimized for train data, losing its generality, the model parameters were updated only when the MSE loss of the validation data was reduced at every epoch. The model was trained over 301 epochs.

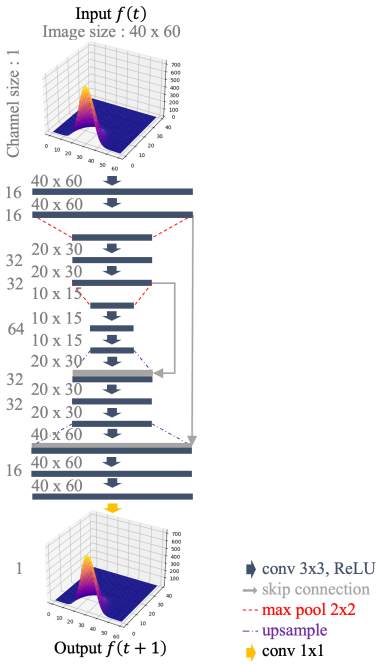

Fig. 1. Overview of our DL-based FPL solver model. Conv 3x3 stands for a convolutional layer with 3x3 kernel and padding with 1 pixel. Skip connection copies and concatenates feature maps in the encoder and upsampled feature maps in the decoder. In the encoder, image size is halved by 2x2 max pooling and in the decoder, it's upsampled to its original size.

2.4 Model evaluation

To evaluate our model, the test dataset which is not covered in the training process was used (Sec. 2.2). For evaluation metric, we used Peak Signal-to-Noise Ratio (PSNR), defined as

$$PSNR = 10 \cdot \log_{10}\left(\frac{s^2}{MSE}\right), \qquad (3)$$

where s is the difference between each image's maximum and minimum value and MSE is what defined in (Eq. 2).

2.5 Results

Comparison between the output from our DL-based FPL collision operator and the ground truth from prior FVM simulation is presented in Fig. 2. This shows the results of the first time-step on the three different anisotropic temperature condition cases in the test dataset. Table I shows the average PSNR over the entire 49 time-steps of each three cases. In terms of PSNR, all three cases got a high score. Prediction performance for $T_{v\parallel} = 78$eV was best, followed by 88eV and 98eV in order. This implies that our model is more difficult in predicting cases as $T_{v\parallel}$ is closer to $T_{v\perp} = 100$eV, i.e., $T_{v\parallel}$ and $T_{v\perp}$ are so close that pdf change over a time is too small (max ~0.02% difference of the previous time step). However, more meaningful thing is that we found our model's output converged to the equilibrium state with diffusive behavior, which indicates neural networks can learn the physical phenomenon governed by the FPL collision equation.

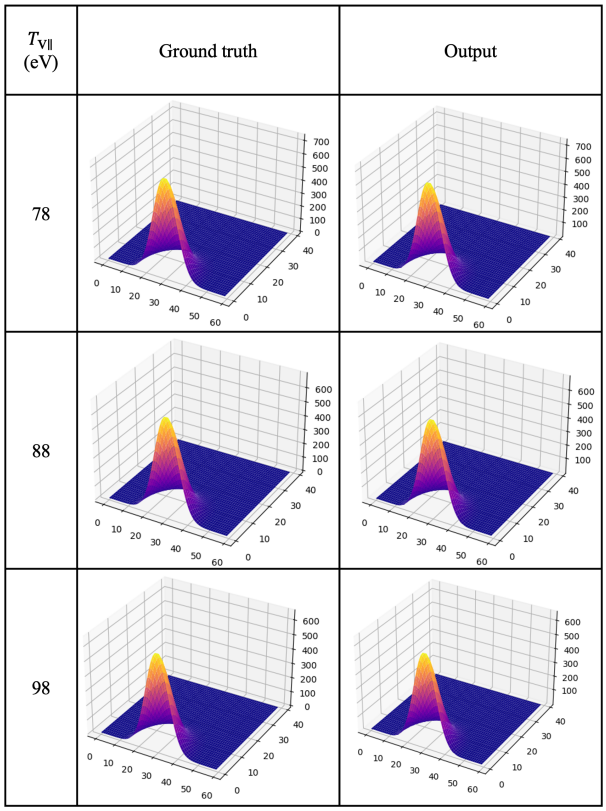

Fig. 2. Comparison between ground truths and DL-based FPL model outputs. Data are from the first time-time step of each temperature case.

Table I: PSNR for test cases

$T_{v\parallel}$ (eV)	PSNR
78	57.4380
88	56.8336
98	56.0377

3. Conclusions

We introduced a CNN-based DL-FPL model to accelerate the FPL collision operator. The deep learning model efficiently reproduced results from a FVM based FPL solved, with high PSNR. Replacing computationally expensive collision operators with deep learning models is anticipated to accelerate computation in future multi-

species simulations. In this preliminary study, our DL model produced a pdf (f) at the next time step. To handle multi species collision later, however, we are currently working on an advanced model that produces a pdf change (Δf). In addition, we are developing a method that our DL model satisfies physical conservation such as mass, momentum, and energy. For training models, we could implement conservation constraints into loss functions, which we will leave as future work.

REFERENCES

[1] A. Dener, M. A. Miller, R. M. Churchill, T. Munson and C. S. Chang, Training neural networks under physical constraints using a stochastic augmented Lagrangian approach, arXiv preprint arXiv:2009.07330, 2020.
[2] E. S. Yoon and C. S. Chang, A Fokker-Planck-Landau collision equation solver on two-dimensional velocity grid and its application to particle-in-cell simulation, Physics of Plasmas, Vol. 21, 032503, 2014.
[3] Ronneberger O., Fischer P. and Brox T., U-Net: Convolutional Networks for Biomedical Image Segmentation, In International Conference on Medical image computing and computer-assisted intervention, Springer, pp. 234-241, 2015.