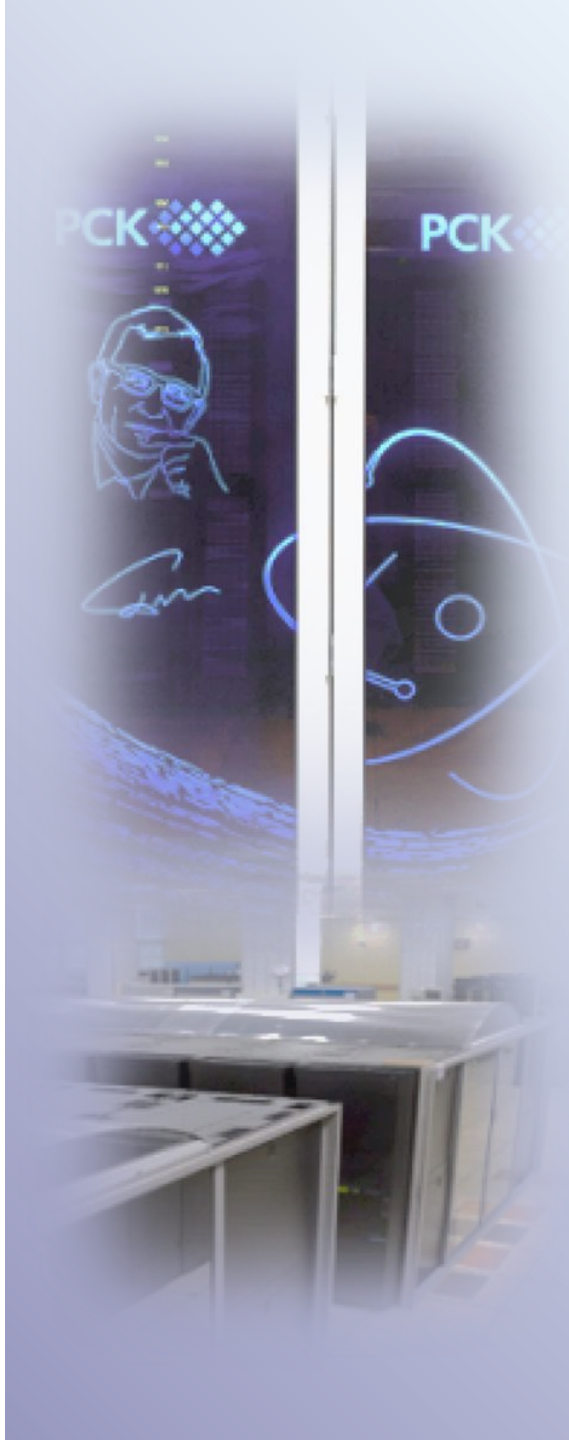
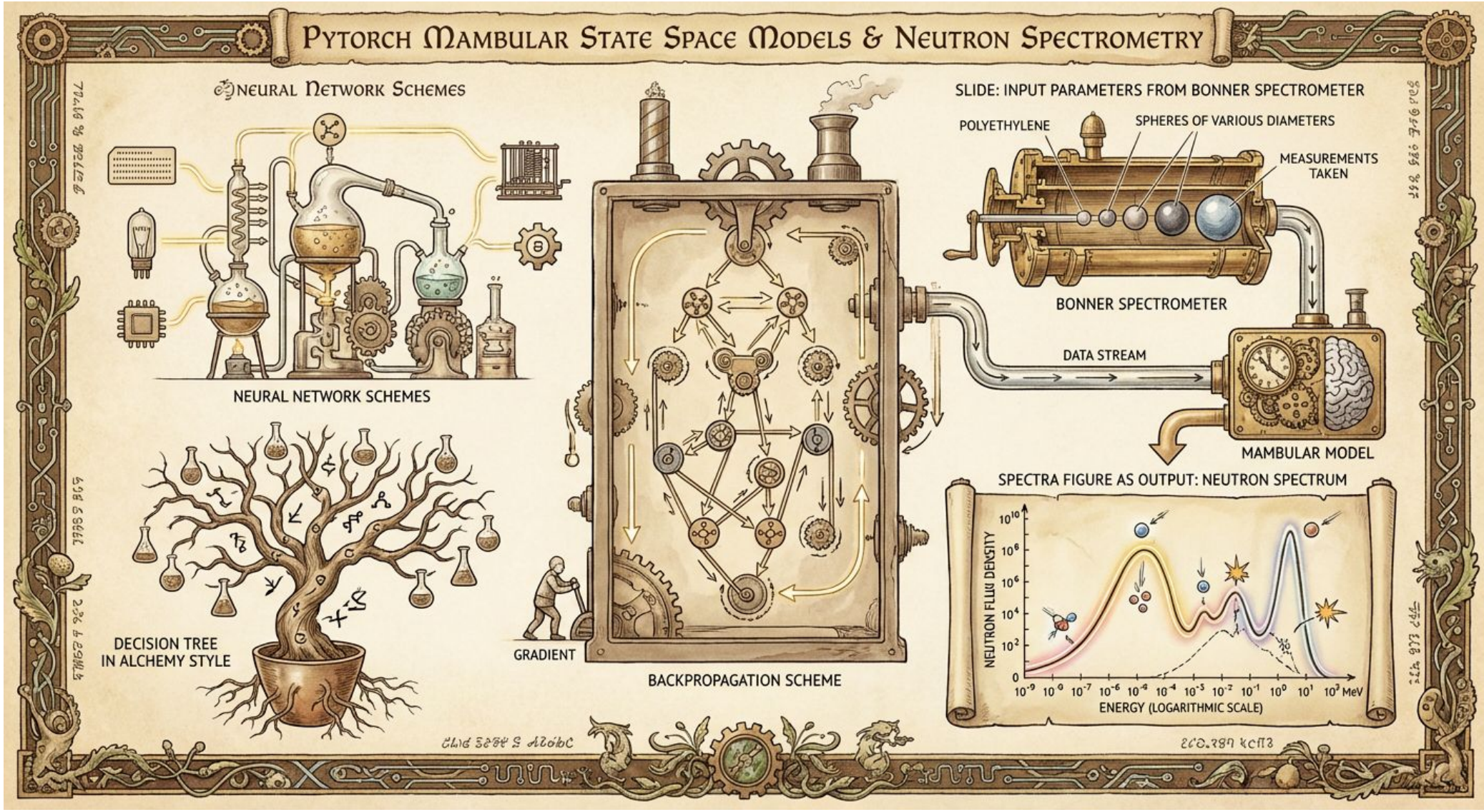
Training neural networks in PyTorch on HybriLIT GPUs

Chizhov Konstantin, PhD
senior researcher
Joint Institute for Nuclear Research,
Meshcheryakov Laboratory of Information Technologies, Dubna, Russia

HybriLIT WORKSHOP 2025: Towards Efficient Scientific Computing (25-26.11.2025)



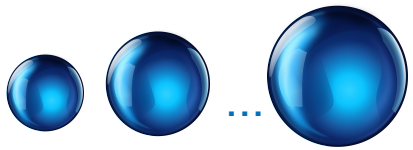
Methodology



Stack of technologies

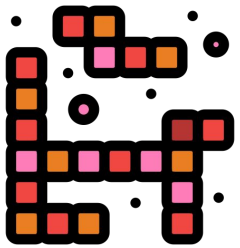
Advanced deep learning architecture for tabular data

10 Measurements

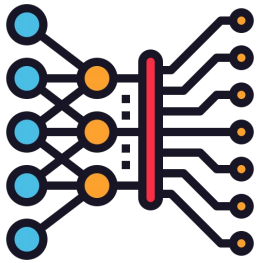


^{6}LiI(Eu) detector
GSF[1]

Estimator



min-max scaler,
peicewise linear encoding
(PLE)[2], dimensions d = 20

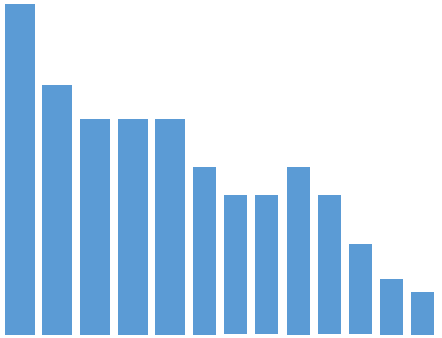


Mambular, TabM, ResNet,
FTTransformer, MLP, SAINT,
TabulaRNN, NDTF, NODE[2]...

	Name	Type	Params	Mode
0	loss_fct	MSELoss	0	train
1	estimator	Mambular	315 K	train
2	estimator.embedding_layer	EmbeddingLayer	12.8 K	train
3	estimator.mamba	Mamba	302 K	train
4	estimator.tabular_head	MLPhead	65	train

315 K **Trainable params**
0 Non-trainable params
315 K Total params
1.261 Total estimated model params size (MB)
69 Modules in train mode
0 Modules in eval mode

60 Targets



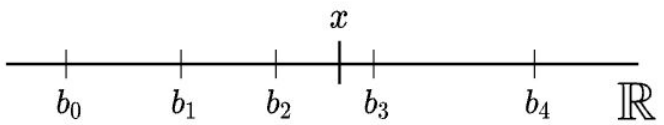
$E_1 \ldots E_S$
$S = 60$

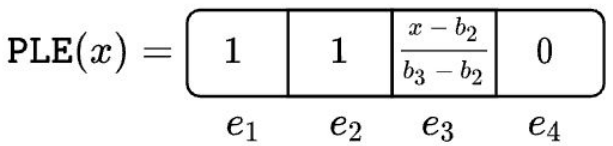
Feature transformations and processing in Mambular

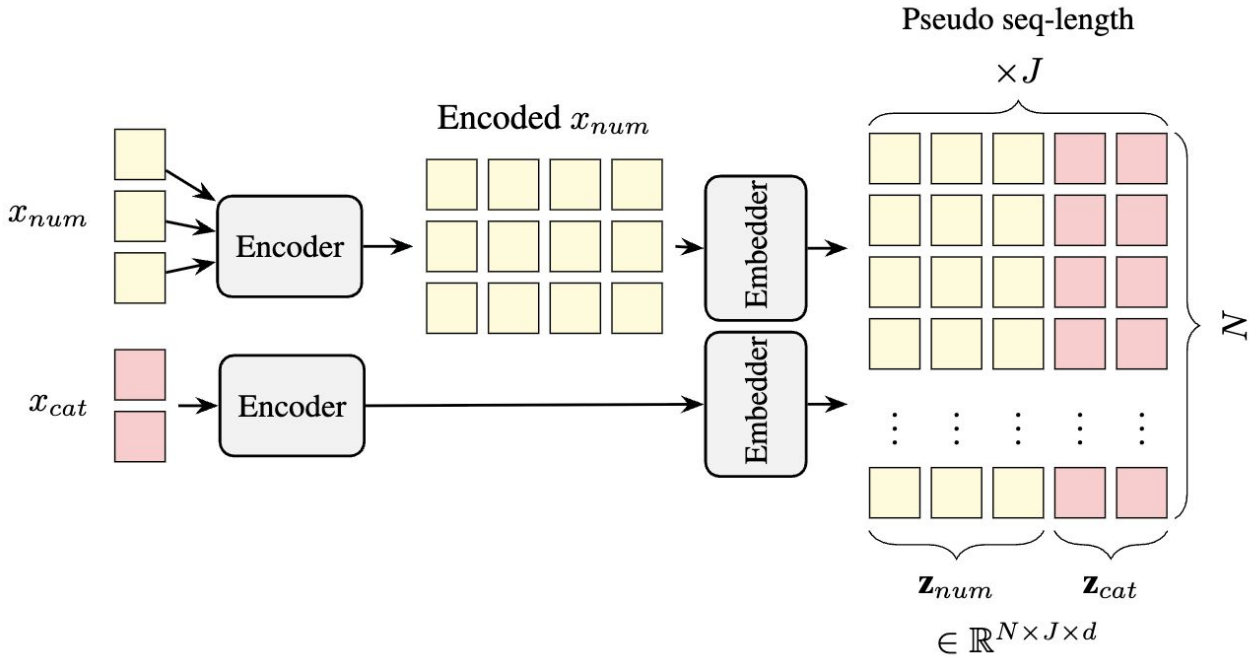
- Each numerical feature (*x*) is encoded with *PLE* before being passed through the linear layer for rescaling.
- *Decision trees* are used for detecting the bin boundaries, b_t,
- *Embeddings* (e_t) are passed jointly through a stack of *Mamba layers*.

$$\mathbf{PLE}(x) = [e_1, \ldots, e_T] \in \mathbb{R}^T$$

$$e_t = \begin{cases} 0, & x < b_{t-1} \ \mathbf{AND} \ t > 1 \\ 1, & x \geq b_t \ \mathbf{AND} \ t < T \\ \frac{x - b_{t-1}}{b_t - b_{t-1}}, & \text{otherwise} \end{cases}$$





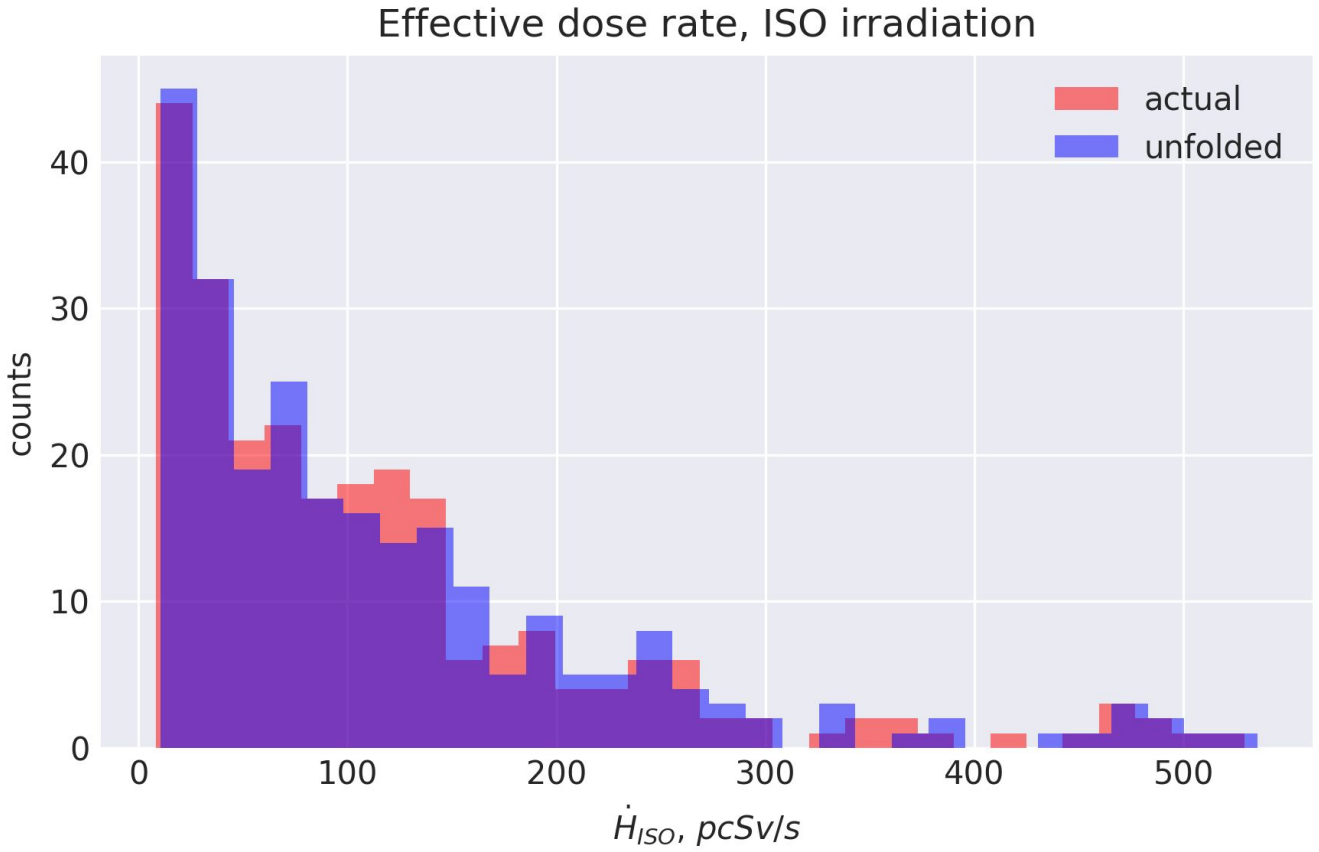


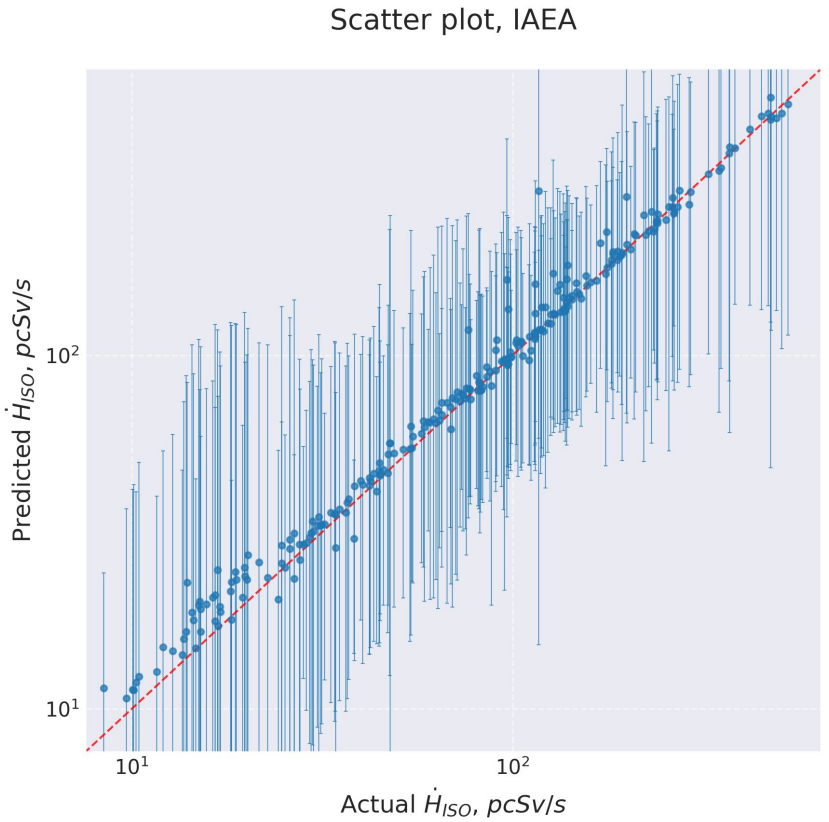
Gorishniy, Y., Rubachev, I., and Babenko, A. (2022). On embeddings for numerical features in tabular deep learning. Advances in Neural Information Processing Systems, 35:24991–25004.

Models for tabular data

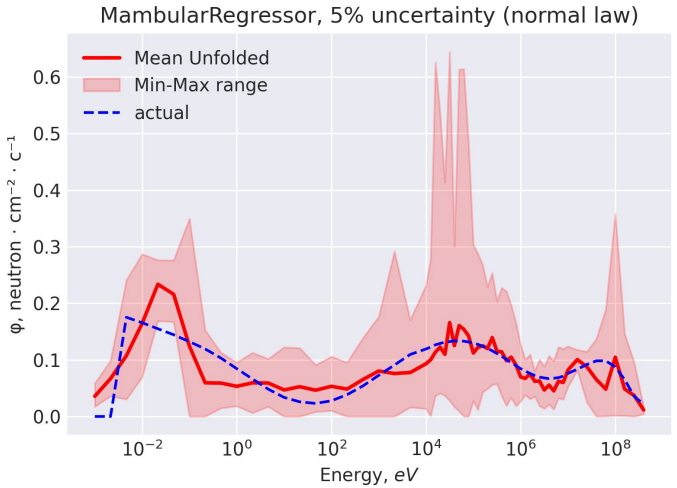
Model	Description
Mambular	A sequential model using Mamba blocks specifically designed for various tabular data tasks (arXiv:2408.06291).
TabM	Batch Ensembling for a MLP(Gorishniy et al.).
MLP	A classical Multi-Layer Perceptron (MLP) model for handling tabular data tasks (arXiv:2408.06291).
FTTransformer	Feature Tokenizer + Transformer. A model leveraging transformer encoders for tabular data (Gorishniy et al.).
NODE	Neural Oblivious Decision Ensembles (Popov et al.).
ResNet	An adaptation of the ResNet architecture for tabular data applications (Gorishniy et al 2021).
TabTransformer	A transformer-based model for tabular data, enhancing feature learning capabilities (Huang et al.).
MambaTab	A tabular model using a Mamba-Block on a joint input representation. Not a sequential model. (arXiv:2401.08867).
TabulaRNN	A Recurrent Neural Network for Tabular data (arXiv:2411.17207v).
MambAttention	A combination between Mamba and Transformers (arXiv:2411.17207v1).
NDTF	A neural decision forest using soft decision trees (Kontschieder et al.)
SAINT	Improve neural networks via Row Attention and Contrastive Pre-Training (arXiv:2106.01342v1).

Mambular results. Dose assessment for 375 real cases

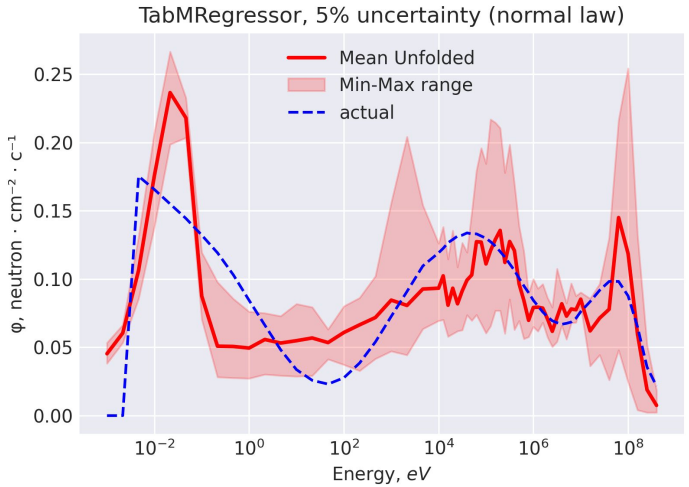




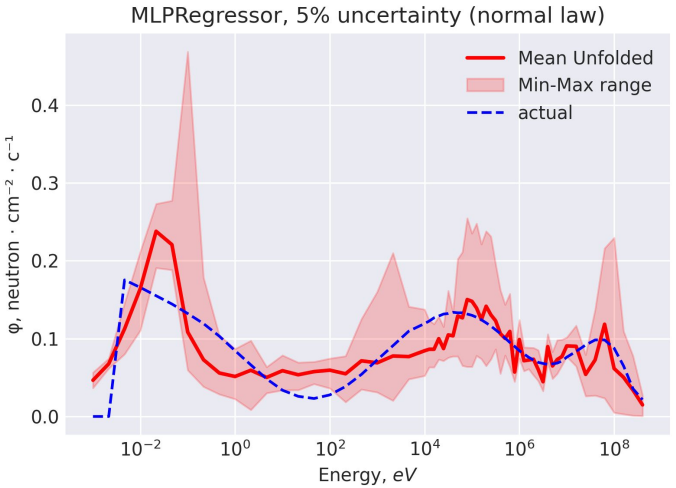
Comparison of models for JINR phasotron hard field



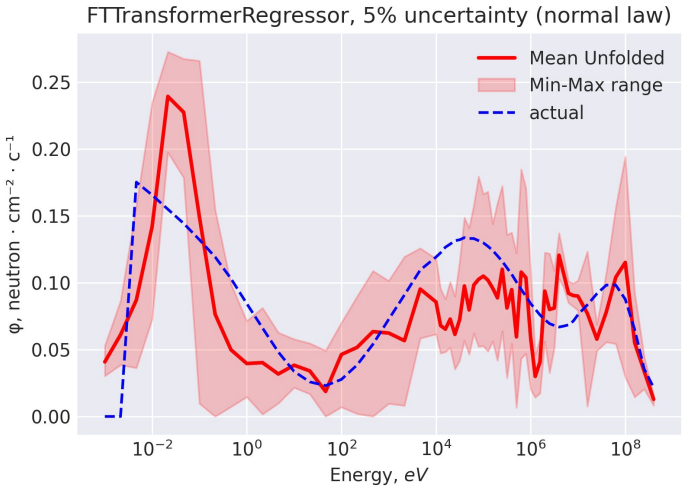
Mambular



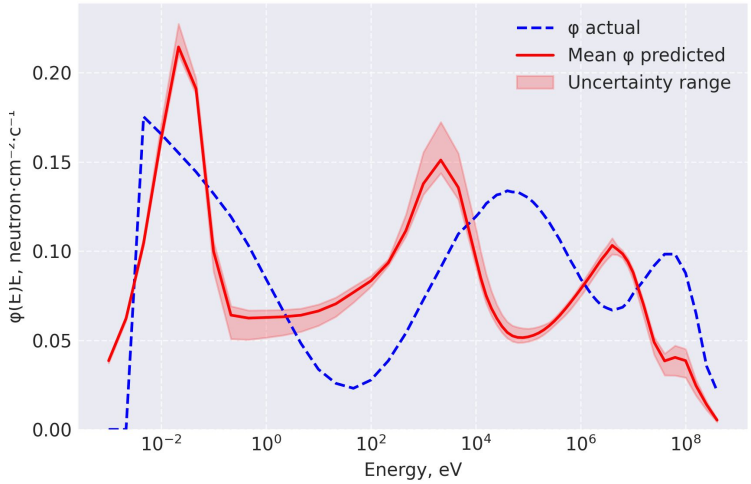
TabM



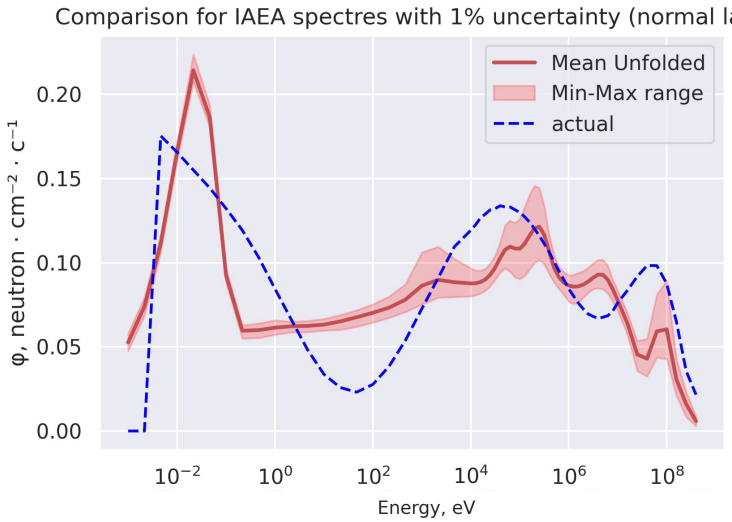
MLP



FTTransformer



LightAutoML



FEDOT

Comparison of models

Mean metrics for the test dataset.

	№	Model name	R^2	MSE	MAE	Cosine similarity	Pearson correlation	MMD	Wasserstein distance	MAPE for dose rate, %
DL	1	Mambular	0,803	3,88E-04	5,84E-03	0,937	0,903	0,043	3,80E-03	9,42
	2	FT Transformer	0,713	7,58E-04	7,94E-03	0,908	0,858	0,043	5,11E-03	14
	3	TabM	0,742	5,60E-04	7,05E-03	0,917	0,871	0,097	4,77E-03	15,1
	4	MLP	0,755	4,98E-04	6,67E-03	0,923	0,879	0,097	4,53E-03	14,54
	5	ResNet	0,755	5,86E-04	7,00E-03	0,921	0,881	0,045	4,71E-03	13,41
	6	TabulaRNN	0,779	3,66E-04	6,14E-03	0,929	0,892	0,043	4,03E-03	12,27
	7	NODE	0,779	5,04E-04	6,36E-03	0,930	0,893	0,034	4,18E-03	9,5
	8	SAINT	0,683	8,31E-04	7,81E-03	0,896	0,848	0,930	5,22E-03	132,2
	9	NDTF	0,711	0,001	0,008	0,909	0,859	0,052	4,90E-03	11
AutoML	10	LightAutoML	0,740	3,95E-04	6,76E-03	0,916	0,866	0,087	4,51E-03	14
	11	FEDOT	0,750	3,55E-04	6,34E-03	0,920	0,873	0,055	4,13E-03	9,15

On a regular basis, the capabilities of nuclear enterprises around the world are tested through international exercises and must achieve within 30% error for neutron dose measurements, https://doi.org/10.1080/00295639.2025.2458437.

SLURM


HAPPY DWARVES
rnostly foll pouriung the knowledge of engopy-operation.
carling SSU systems syteminem against a modern dücziven level.
KIND ELVES
use piicsent here the principles that in-ventionary to-pod poiit êlexistlines.
#!/bin/bash
#SBATCH --job-name=ALCHEMICA
#SBATCH --output=transmute.out
#SBATCH --ntasks=4
srun python3 elixir_sim.py
HAPPY DWARVES
pouring reoil from gloweo or.doying insinehients.
KIND ELVES
purvs amhercy with calcbration of ERF- Eljvsie elvets
$ sbatch ALCHEMICA.sh
cunuies in a iepered pwple of spaices
srun python3 elixir_sim.ion to your happineding elows

Setting the environment

Install packages:

to the user virtual environment

- module add Python/v3.10.13
- pip install pytorch -U

Commands for terminal

- module avail - check available modules
- sinfo - view information about Slurm nodes and partitions.
- sbatch **SLURMtest.sh**
- squeue - view information about jobs located in the Slurm scheduling queue
- scancel - kill the task

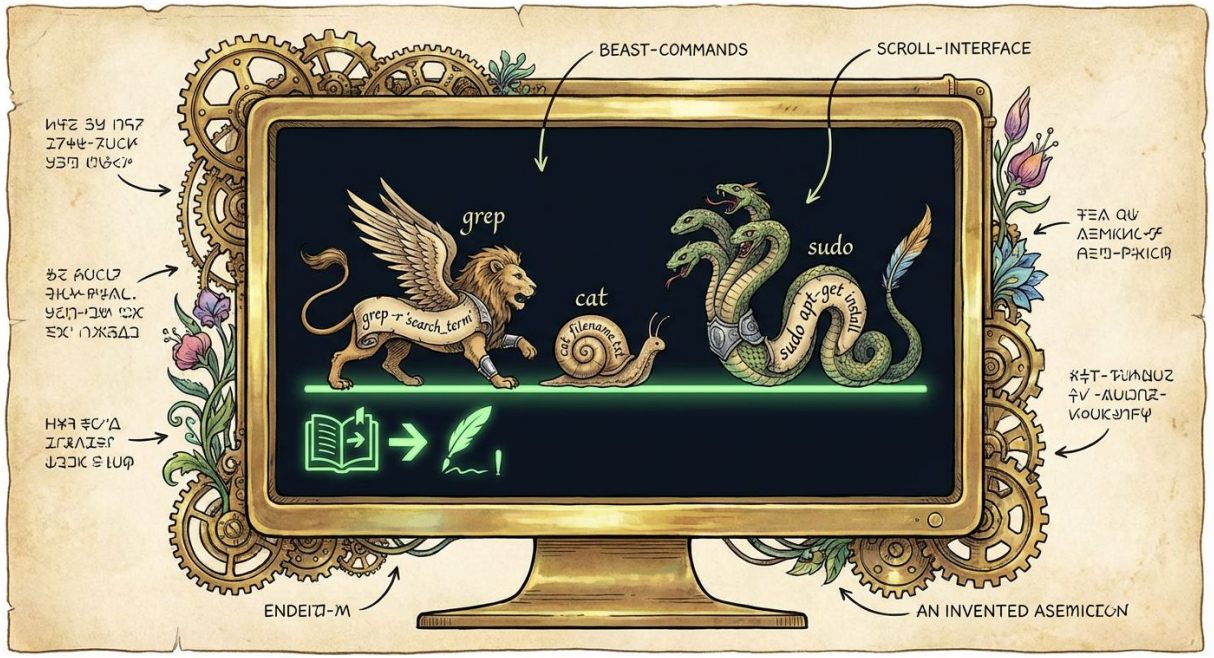
SLURMtest.sh

```
#!/bin/bash
#SBATCH --mem=0                 #request all the memory on a node
#SBATCH -p dgx                  # available from sinfo
#SBATCH --gres=gpu:1            # number of GPUs per node
#SBATCH -n 50                   # число используемых процессов
#SBATCH -t 14-00:00:0           # 14 days (maximum)

module add Python/v3.10.13
module add cuda/v12.4

module list  # active modules
python -V    # check the python version
pip freeze   # get a list of python packages

hostname
lscpu          # information about the CPU architecture.
free -h        #available — this is the real “free” memory for applications

python3 testGPU.py
```

Set the environment

VENV with pip:

```
python3 -m venv .venv
source .venv/bin/activate
which python
pip install -r requirements.txt
python3 testGPU.py
deactivate
```



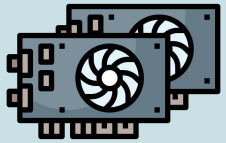
VENV with poetry:

```
module add Python/v3.10.13
pip install poetry -U
poetry init
poetry add scikit-learn scipy numpy
poetry install
poetry env use python3.10.13
poetry shell
python3 testGPU.py
```

if initialization of the project

Testing the CUDA availability

testGPU.py



```
import torch

print(torch.cuda.is_available())
print(torch.cuda.device_count())
print(torch.cuda.get_device_name(0))
```

```
True
1
Tesla V100-SXM2-16GB
```

Simple neural network

pytorch lightning

Model
- Defining the architecture
- Training step
- Optimizer

Data
- Loading
- Transforming
- Breaking into batches

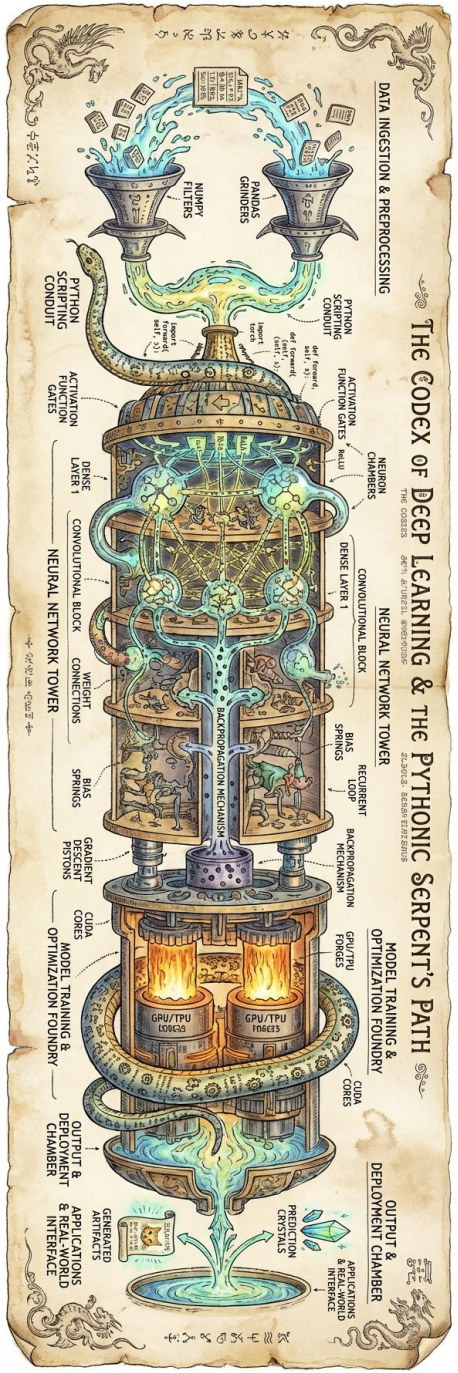
Training

```
import pytorch_lightning as pl
import torch
import torch.nn.functional as F
from torch.utils.data import DataLoader, TensorDataset

class SimpleNN(pl.LightningModule):
    def __init__(self, input_dim=10, hidden_dim=50, output_dim=1):
        super(SimpleNN, self).__init__()
        self.fc1 = torch.nn.Linear(input_dim, hidden_dim)
        self.fc2 = torch.nn.Linear(hidden_dim, output_dim)

    def forward(self, x):
        x = F.relu(self.fc1(x))
        x = self.fc2(x)
        return x

    def training_step(self, batch, batch_idx):
        x, y = batch
        y_hat = self(x)
        loss = F.mse_loss(y_hat, y)
        self.log('train_loss', loss)
        return loss

    def configure_optimizers(self):
        return torch.optim.Adam(self.parameters(), lr=0.001)

model = SimpleNN(10, 50, 1)

trainer = pl.Trainer(gpus=-1, accelerator='gpu', strategy='ddp')
trainer.fit(model, train_loader)
```

Mambular framework

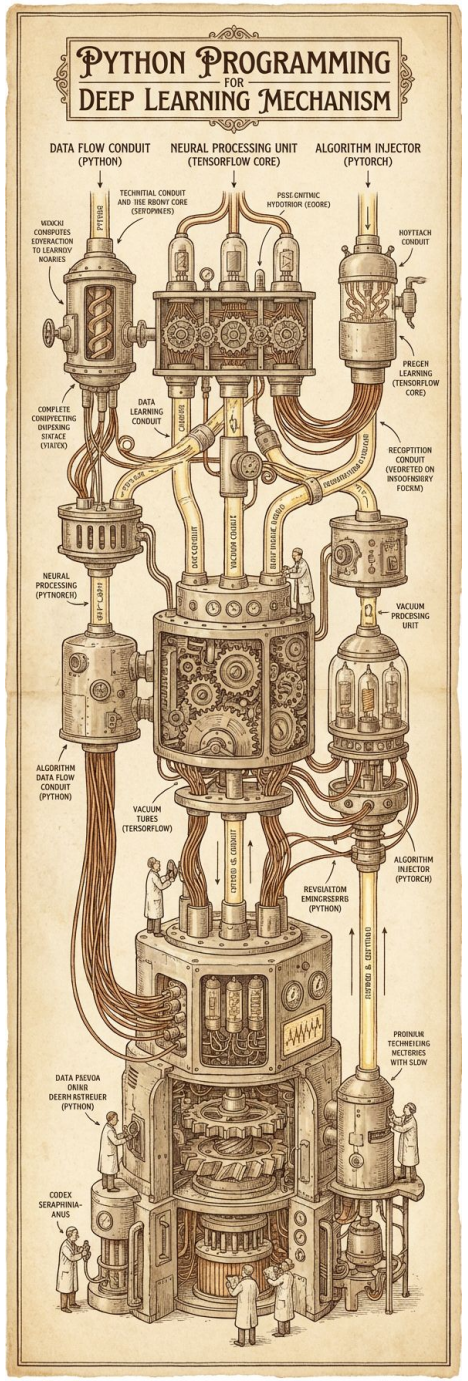
definition of the model is similar to the scikit-learn style

```
def train_regressor(
        X_train,
        y_train,
        model_class: Union[
        "MambularRegressor",
        "TabMRegressor",
        "ResNetRegressor",
        "FTTransformerRegressor",
        "TabulaRNNRegressor",
        "MLPRegressor",
        "NODERegressor",
        "NDTFRegressor",
        "SAINTRegressor",
        ],
        max_epochs: int,
        lr: float,
        rundir: str = "./results",
        verbose: bool = True,
):
```



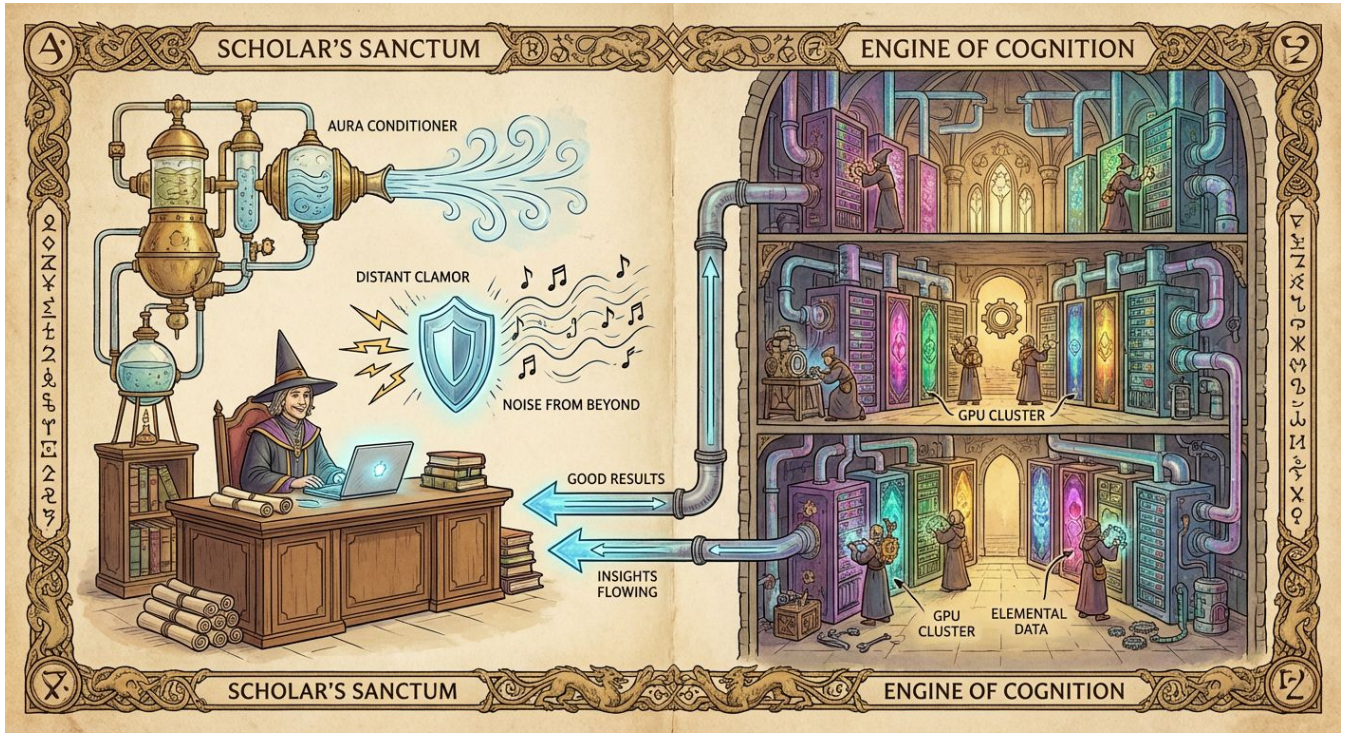
Mambular framework

```
regressor = model_class()

regressor.fit(
        X_train,
        y_train,
        max_epochs=max_epochs,
        lr=lr,
        checkpoint_path=os.path.join(
                rundir, f"model_checkpoints_{model_class.__name__}"
        ),
        log_every_n_steps=100,
        enable_progress_bar=False,
        logger=csv_logger,
        dataloader_kwargs={"num_workers": 8},
        )


    test_prediction = predict_test(models,
    X_test=X_test, verbose=verbose)
```



Conclusions

- It is recommended to set up an independent virtual environment for each of your projects.
- Research what the classes and packages you use are based on.
- Accelerate computations using the GPU's capabilities by running calculations on a remote server.
- Use task management via SLURM.



Thank you!

Publications:

1. **Chizhov K.A. , Bely A. Neutron spectrum unfolding using deep learning models for tabular data, Moscow University Physics Bulletin. Physics series, 2025.**
2. **Chizhov K, Bely A, Starikovskaya M, Automated machine learning spectrum unfolding for neutron spectrometry with Bonner spheres, proceedings of the International Conference "Distributed Computing and Grid-technologies in Science and Education".**
3. Chizhov K.A. et al. Unfolding of the energy spectrum of the neutron radiation flux using the random forest machine learning algorithm. *Modern information technologies and IT education*, 20, 4 (Dec. 2024). ISSN 2411-1473 (in Russian).
4. Chizhov, K., Beskrovnaya, L. & Chizhov, A. Neutron Spectrum Unfolding Method Based on Shifted Legendre Polynomials, Its Application to the IREN Facility. *Phys. Part. Nuclei Lett.* 22, 337–340 (2025). https://doi.org/10.1134/S154747712470239X
5. Chizhov, K., L. Beskrovnaya, and A. Chizhov. "Neutron Spectra Unfolding from Bonner Spectrometer Readings by the Regularization Method Using the Legendre Polynomials." *Physics of Particles and Nuclei* 55.3 (2024): 532-534.
6. Chizhov K, Chizhov A. Dose assessment of personnel neutron irradiation on high-energy accelerators using a multi-sphere Bonner spectrometer. *Mathematical Modeling* 7 (2), 63-64 (2023).

kchizhov@jinr.ru