Robust Deep Learning

Alexey Boldyrev, Andrey Shevelev, Fedor Ratnikov

FCS Scientific Conference
HSE Voronovo 27/10/2025

Image credit: ‘Glacier du Rhone au haut du Valais’
by Claude Niquet after Jean Séraphin Désiré Besson
https://wellcomecollection.org/works/e3y95vtv

When Robust Models Are Needed

- Presence of outliers or data errors;
- Under uncertainty and incomplete data;
- For noisy or real-world data;
- In high-stakes domains:
  - The control of the system by a human is difficult;
  - The number of function evaluation is limited;
    - Its cost is higher or comparable to the cost of training the model.

Image source: perplexity.ai

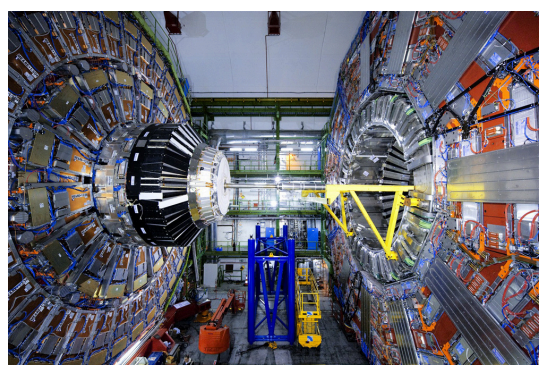


Image credit: CERN

Understanding Robustness in Neural Networks

- The word **robust** is loaded with many (sometimes inconsistent) connotations;
  - Following Huber's definition, robustness signifies insensitivity to small deviations from the assumptions.
- In this study, we look at both outlier resistance and how good a model generalizes from finite training datasets;
  - It is assumed that the training and test samples are randomly and fairly split, so outliers are distributed similarly;
    - It shows their quantity but not the influence on the model.
  - How does the nature and amount of training data affect finding a robust solution?
    - Is there a **minimum training sample size** needed to achieve robust predictions on the test set?

How Do We Define Robustness?

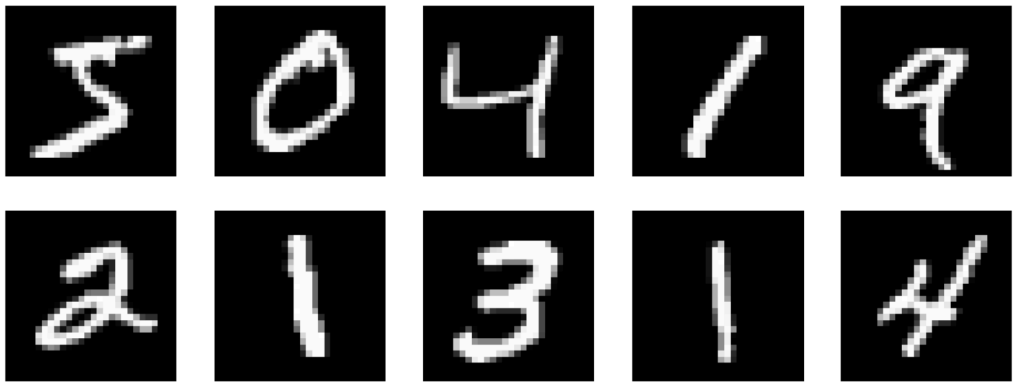
- We propose an empirical definition:
  - A model is called **robust** if the variation in quality among its different instances is minimal.
- What is model instance?
  - All model instances have an identical set of hyperparameters but differ in their:
    - (internal) nondeterministic initial states and algorithms;
    - (external) training samples drawn from the same population.
- C1. By the definition above, any constant model would be absolutely robust.
  - Therefore, to find a practically useful robust model, we need to solve a dual optimization problem: to identify a robust model with the best average performance.
- C2. The robustness of a model depends on the robustness of the quality metric used.

Factors Affecting the Robustness of ML (DL) Model

- Differences in training and test samples;
- Intrinsic nondeterminism of the model and its learning process (see Reproducibility in PyTorch);
  - Initialization of model weights;
  - Nondeterministic transformations (torch.use_deterministic_algorithms);
  - Optimizer behavior.

We measure the robustness by an appropriate statistical measure of the set of test losses of **model instances**, when a specified number of iterations is reached or when an early stopping criterion is met.

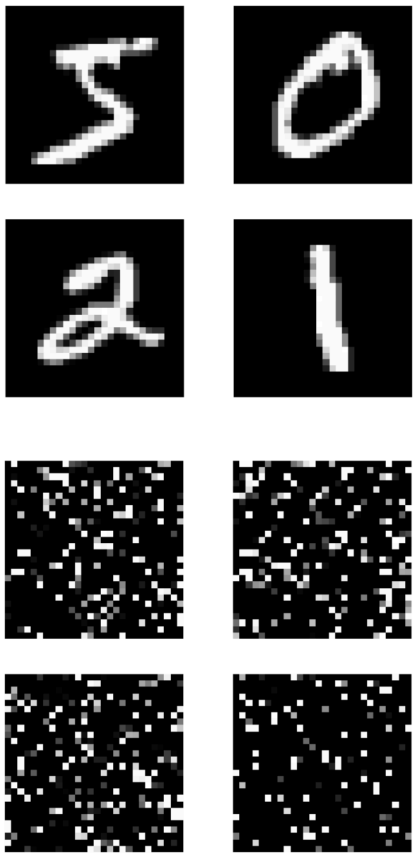
Robustness to Transformations of Data

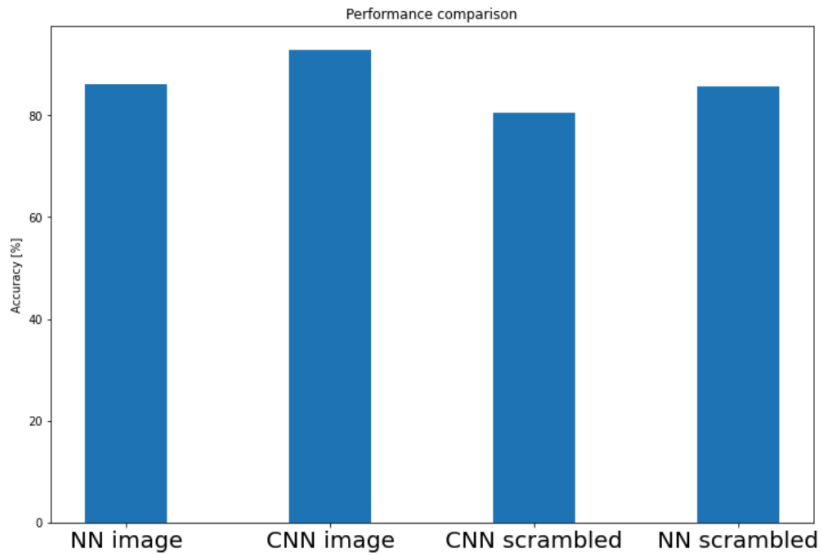


10 classes in MNIST dataset

Convolutional neural networks provide translation invariance:

Robustness to Transformations of Data





Robustness to Transformations of Data

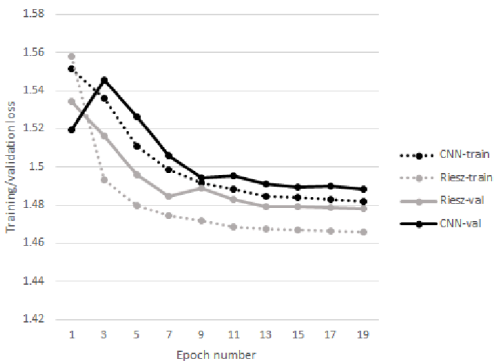
Rotation invariance




Usually achieved by data augmentation

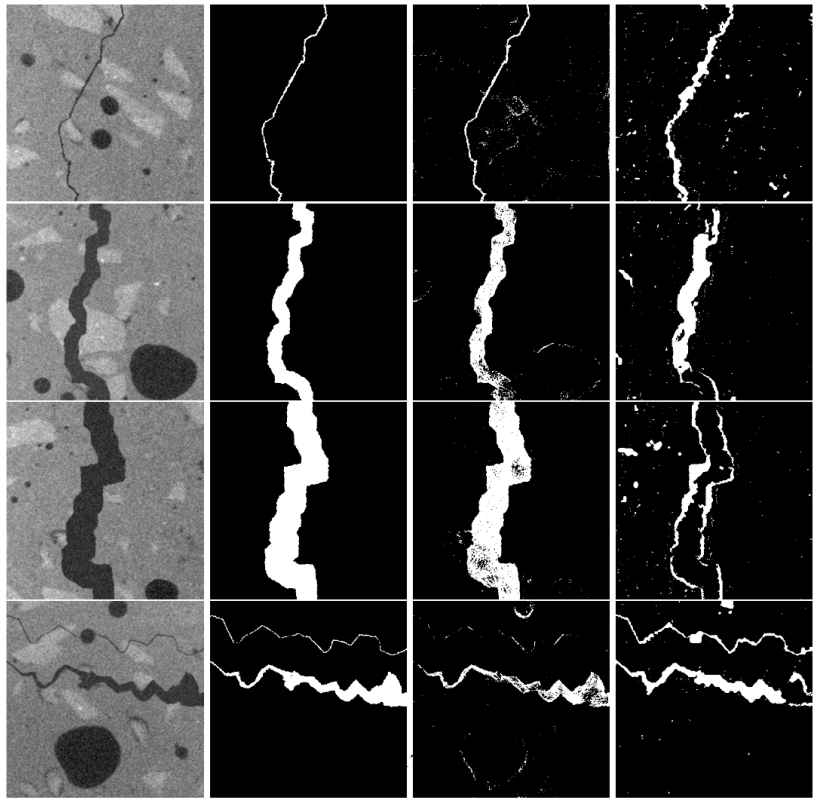
Robustness to Transformations of Data

Scale invariance



Images source: arXiv:2305.04665

Riesz transform neural networks (2024)

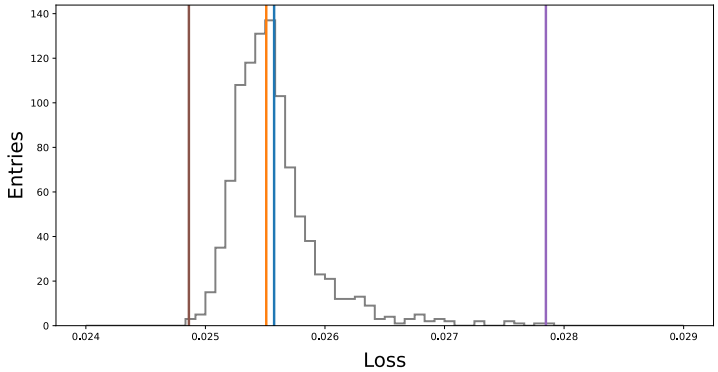


From left to right: input image, ground truth, results of the Riesz network and the U-net with 4 pyramid levels

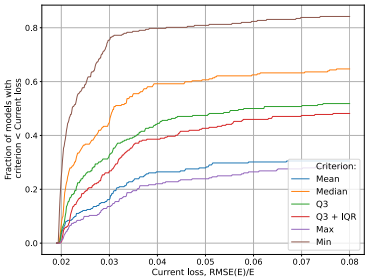
Robust Model Selection Algorithm

Algorithm 1 Model Selection Algorithm

```
Step s = 0
M = Initial set of models
Define selection criterion
Define the number k of instances of each model
while number of models in M ≥ 1 do
    for each model in M do
        Train model instance with new initialization
        Update model robustness value
    end for
    s = s + 1
    if s > k then
        Update selection criterion
        for each model in M do
            if model robustness is worse than
selection criterion then
                Remove model from M
            end if
        end for
    end if
end while
```



Effective reduction of model trainings:



- 41567 (ours)

  vs.

  345600 (full search);
- for 6912 models with

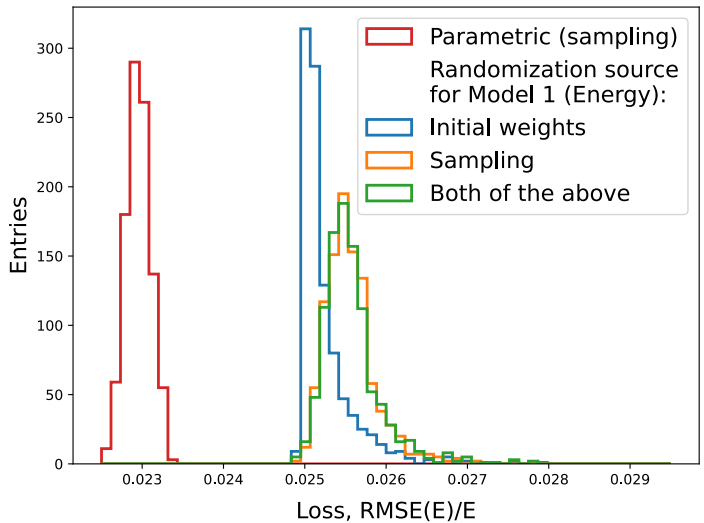
  50 instances each;

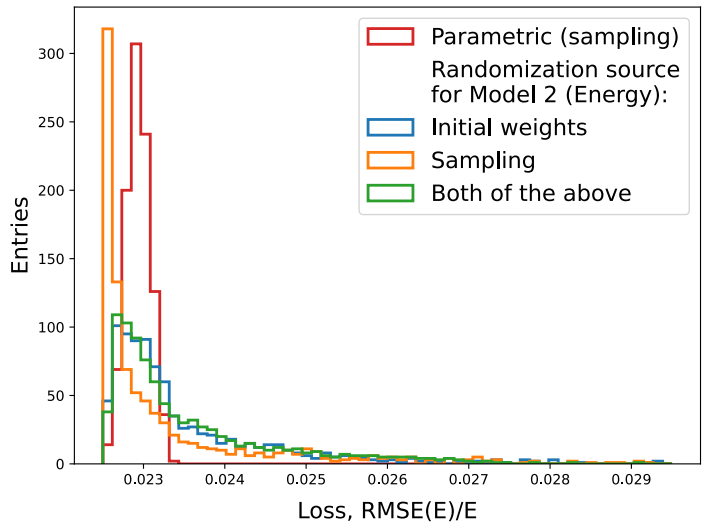
Published in https://doi.org/10.1109/ACCESS.2025.3578926

Results

(Properties of Selected Robust Models)

Impact of Model Nondeterminism on Robustness

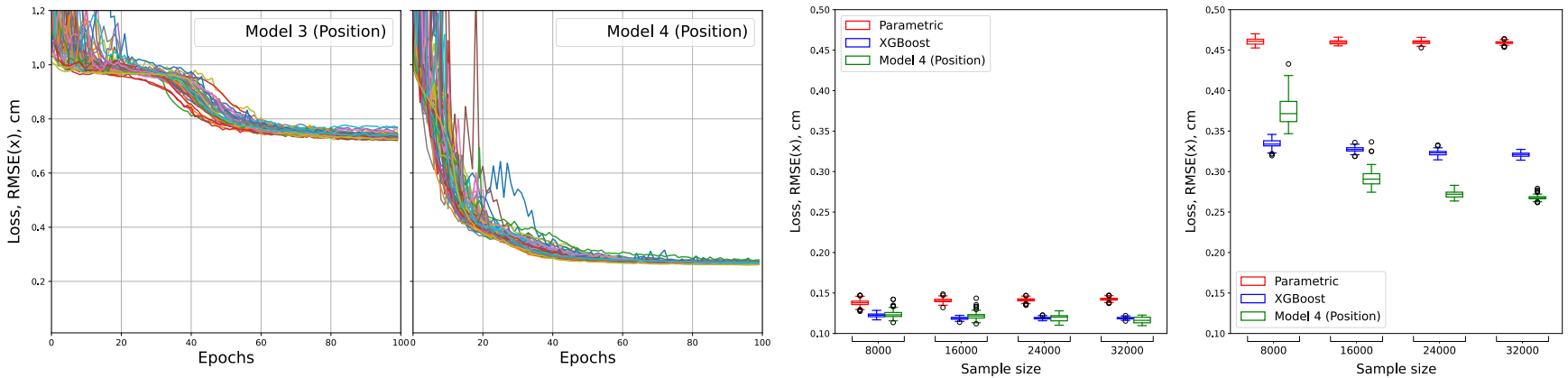




- Kaiming (He) weight initialization is used;
- Each of the 1000 instances of the model is trained

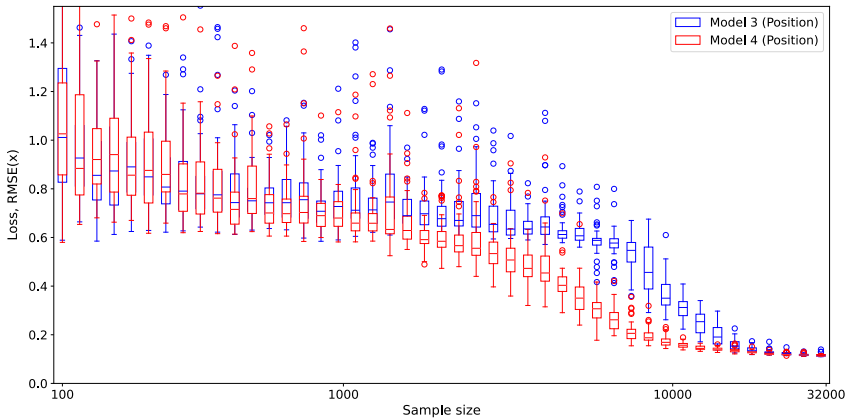
  on a sample of 32k examples randomly drawn from the dataset.

Selected Robust Models for Position Reconstruction



See extra details in backup slide.

Selected Robust Models for Position Reconstruction



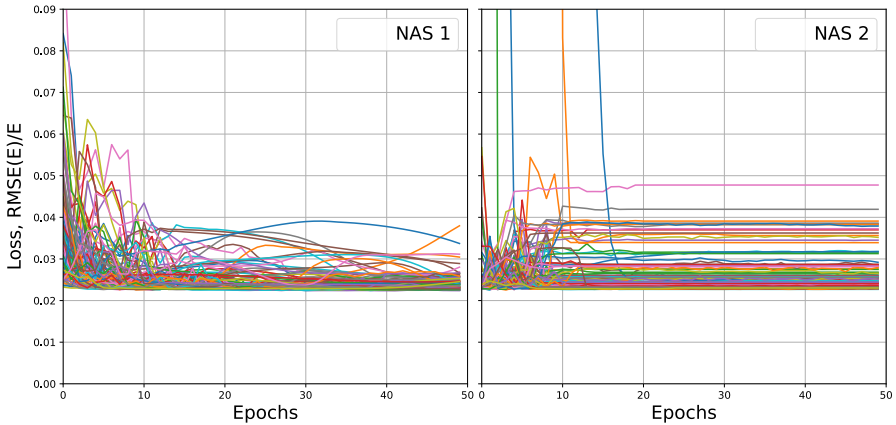
- Simplified dataset;
- Two models:
  - Base;
  - Using inductive biases.

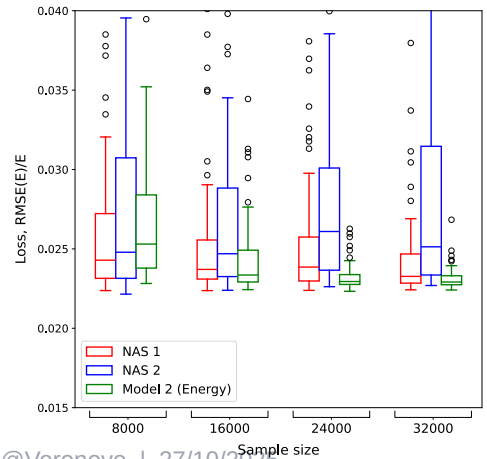
Comparison with NAS approach

- The NAS from the Optuna library is used
- This tool provides the ability to search multiple hyperparameters using the Tree-structured Parzen Estimator algorithm to select hyperparameter values

Model	Activation function	Optimizer	Learning rate	Batch size	Regularization (weight decay)
NAS 1	ReLU	AdamW	0.001	32	0.001
NAS 2			0.01		0.01

```
================================================
Layer                Output Shape        Param #
================================================
NAS 1                [-1]                --
|--Conv2d            [-1, 32, 13, 13]    288
|--ReLU              [-1, 32, 13, 13]    --
|--MaxPool2d         [-1, 32, 6, 6]      --
|--Conv2d            [-1, 16, 4, 4]      4,608
|--ReLU              [-1, 16, 4, 4]      --
|--MaxPool2d         [-1, 16, 1, 1]      --
|--Linear            [-1, 32]            544
|--ReLU              [-1, 32]            --
|--Linear            [-1, 9]             306
|--ReLU              [-1, 9]             --
```

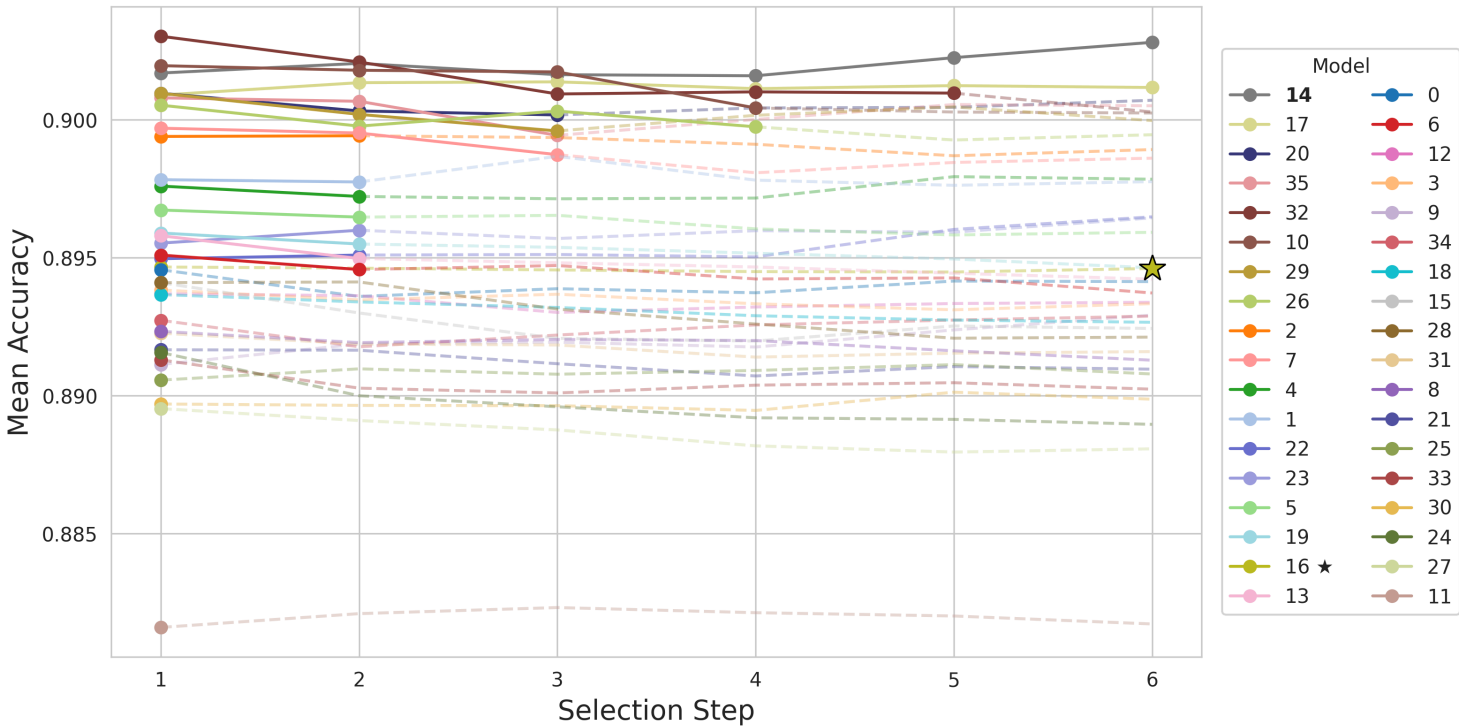




Evaluating Robust Model Selection Algorithm on CIFAR-10

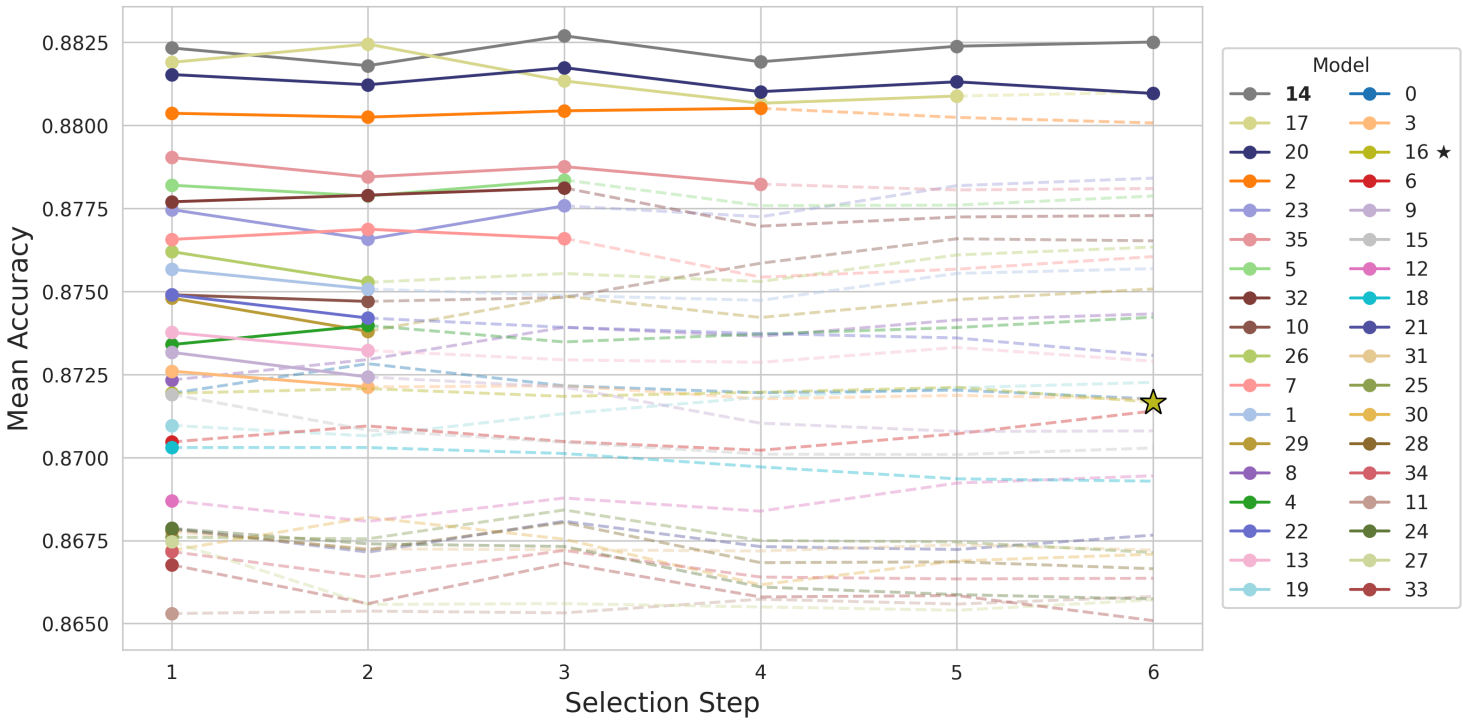
- Dataset **CIFAR-10**:
  - 50k/10k train/validation samples.
- Base model:
  - Benchopt-optimized **ResNet-18** from paperswithcode.com benchmark (archived version);
  - Validation accuracy: 95.55% while trained on augmented sample of 50k examples.
- Model search space generation:
  - Created **36 variations** of the base model by adjusting training hyperparameters:
    - Batch size / Maximum learning rate / L2 regularization parameter.
- Evaluation procedure:

Model Path in the Algorithm: Mean Accuracy



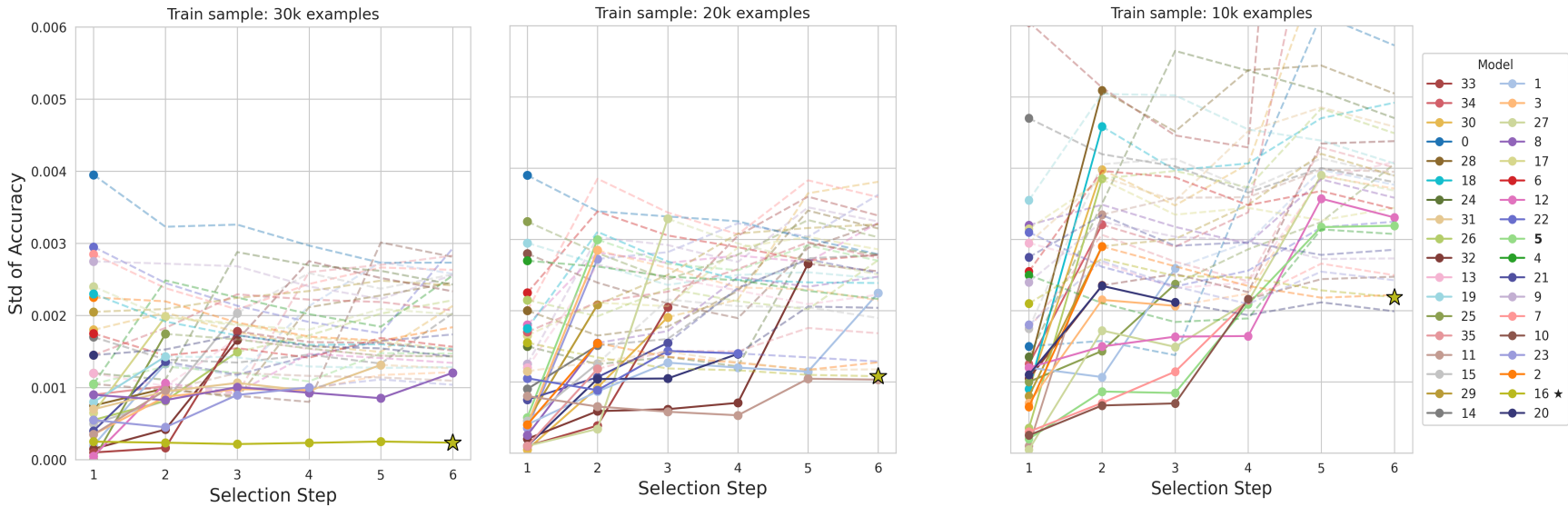
- Train sample: 30k examples
- Warmup steps = 3
- ⭐ - base model

Model Path in the Algorithm: Mean Accuracy



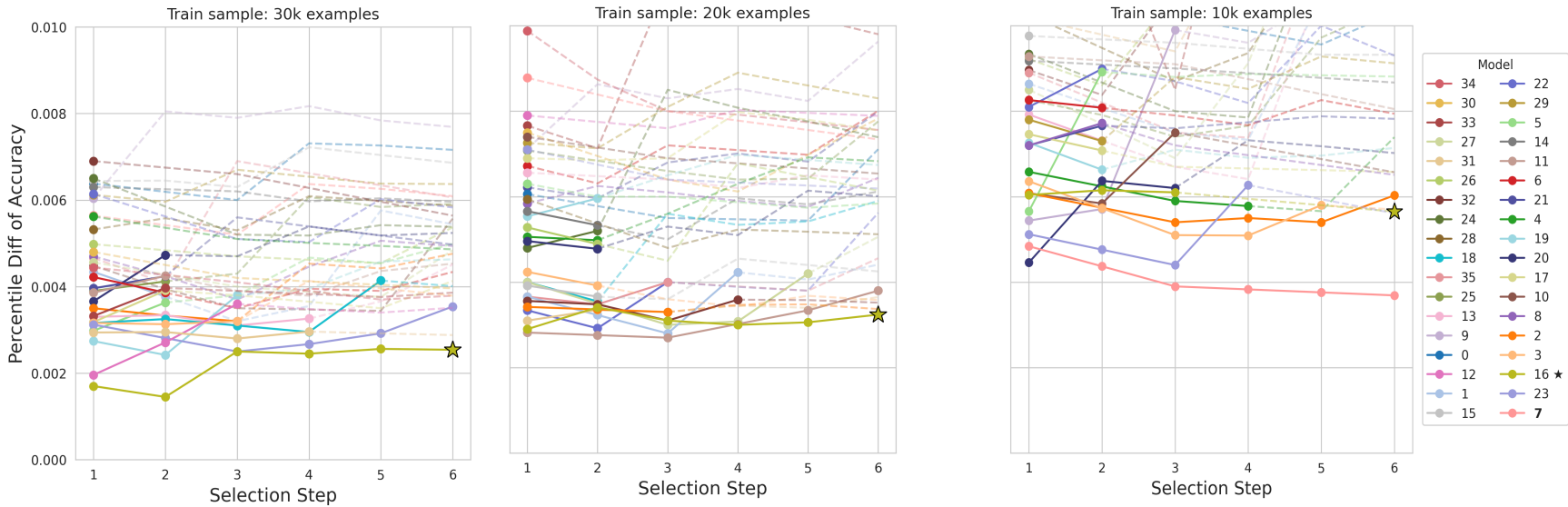
- Train sample: 20k examples
- Warmup steps = 3
- ⭐ - base model

Model Path in the Algorithm: Std of Accuracy



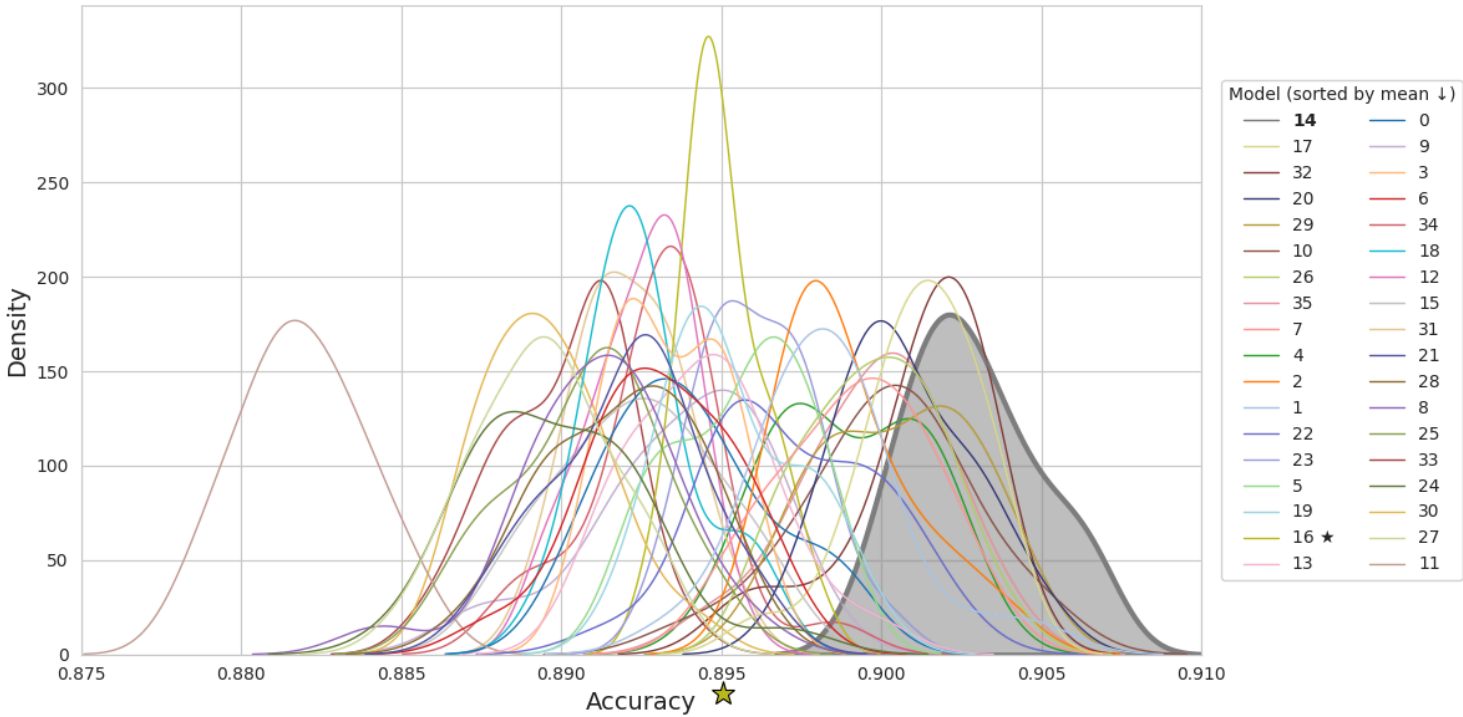
- Warmup steps = 3
- ⭐ - base model

Model Path in the Algorithm: Perc. Diff. of Accuracy



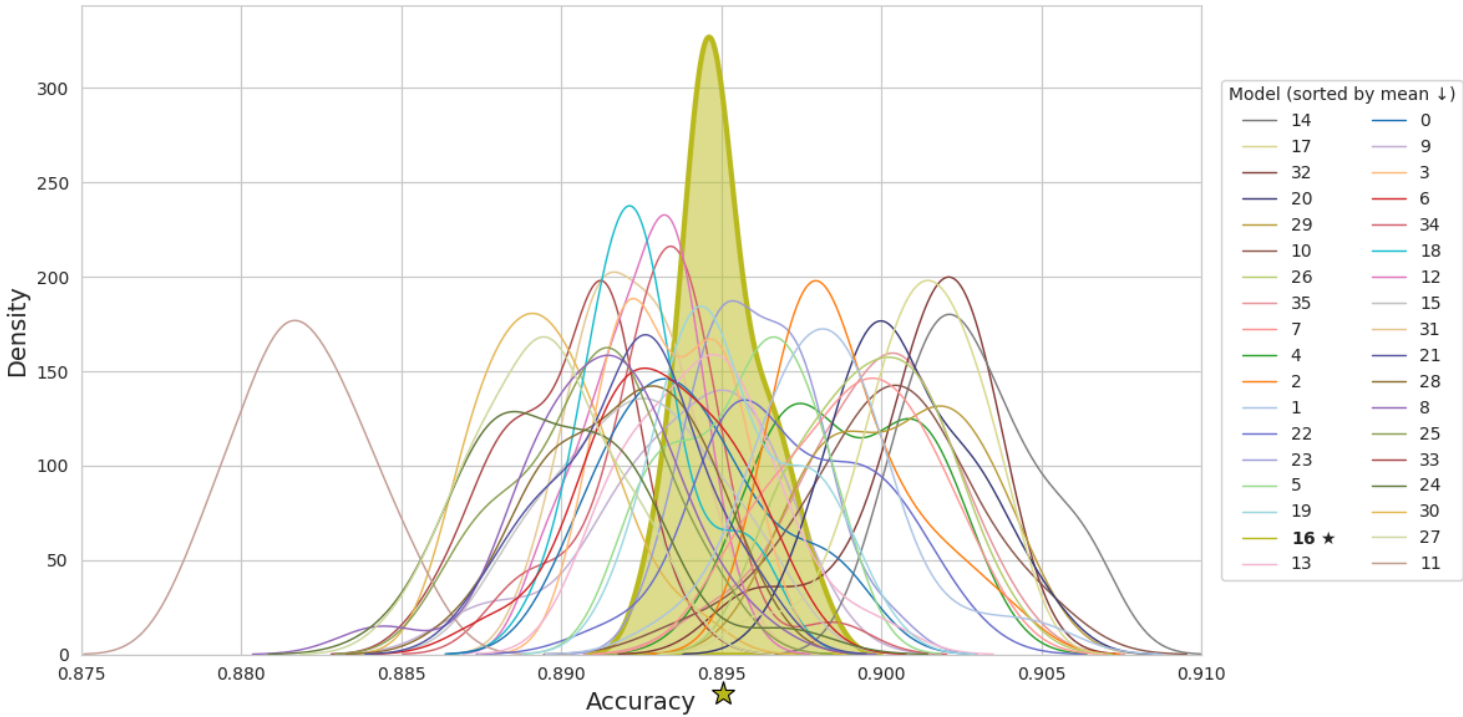
- Warmup steps = 5
- ⭐ - base model
- $PD = Q_{90} - Q_{10}$

Selection Criterion Cross-Check



- Criterion: `mean`
- Train sample: 30k examples
- ⭐ - base model

Selection Criterion Cross-Check



- Criterion: `std`
- Train sample: 30k examples
- ⭐ - base model

Conclusions

- We propose a procedure to measure the robustness of machine learning models.
- We supplement such a procedure with a meta-algorithm for robust model selection.
- The two robust models for two specific problems found using this method have the best convergence and the smallest loss variability among the 2×6912 models considered.
  - The models we found are more robust than the models selected by NAS from a similar search space.
  - ~ 8x speedup in training time is observed compared to an exhaustive search;
    - The total training time can be further reduced by using robust model search on subsamples.

A paper with this method has been published in the IEEE Access journal. DOI: 10.1109/ACCESS.2025.3578926

END

Thank you for the attention!

Happy to answer your questions by e-mail aboldyrev@hse.ru and via Telegram @aboldyrev

Powered by Slidev

Backup slides

Robustness to Transformations of Data

- Translation invariance is achieved in the network architecture (CNN);
- Rotation invariance is achieved by augmenting the training data;
- Scale invariance is achieved in the network architecture (Riesz networks);

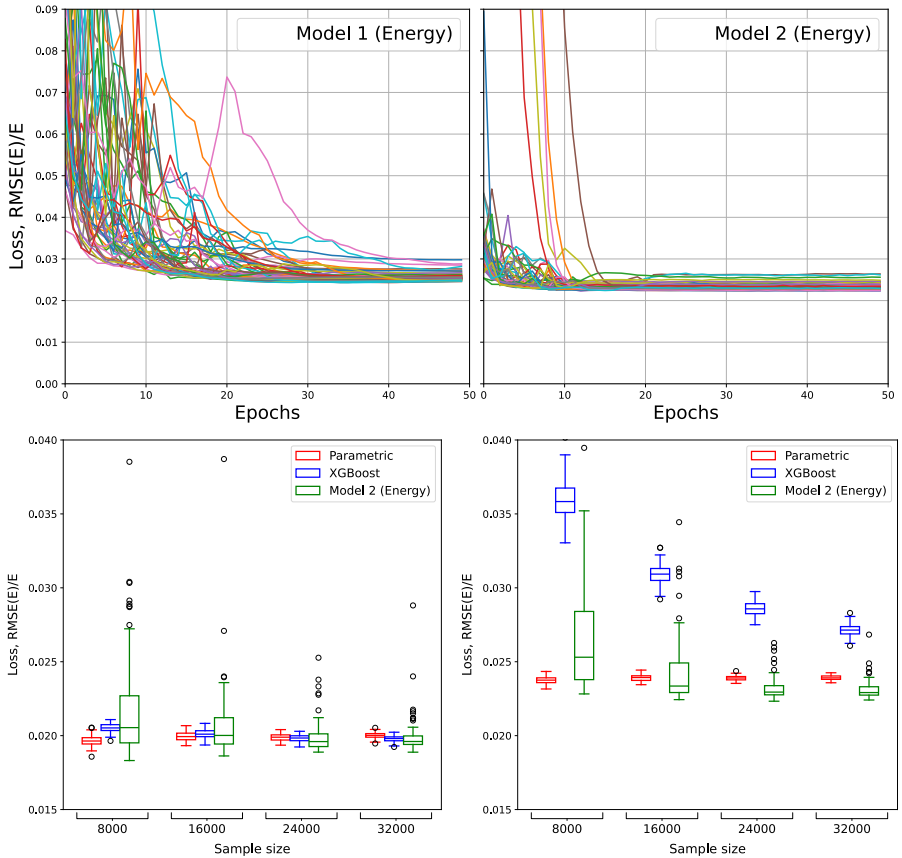
- Is it possible to take the above into account in a capacity of a neural network?
  - What kind of architecture?
- What minimum inductive biases will be sufficient?

The answers to these questions lie in the details of the data and the task.

Selected Models: Energy Reconstruction

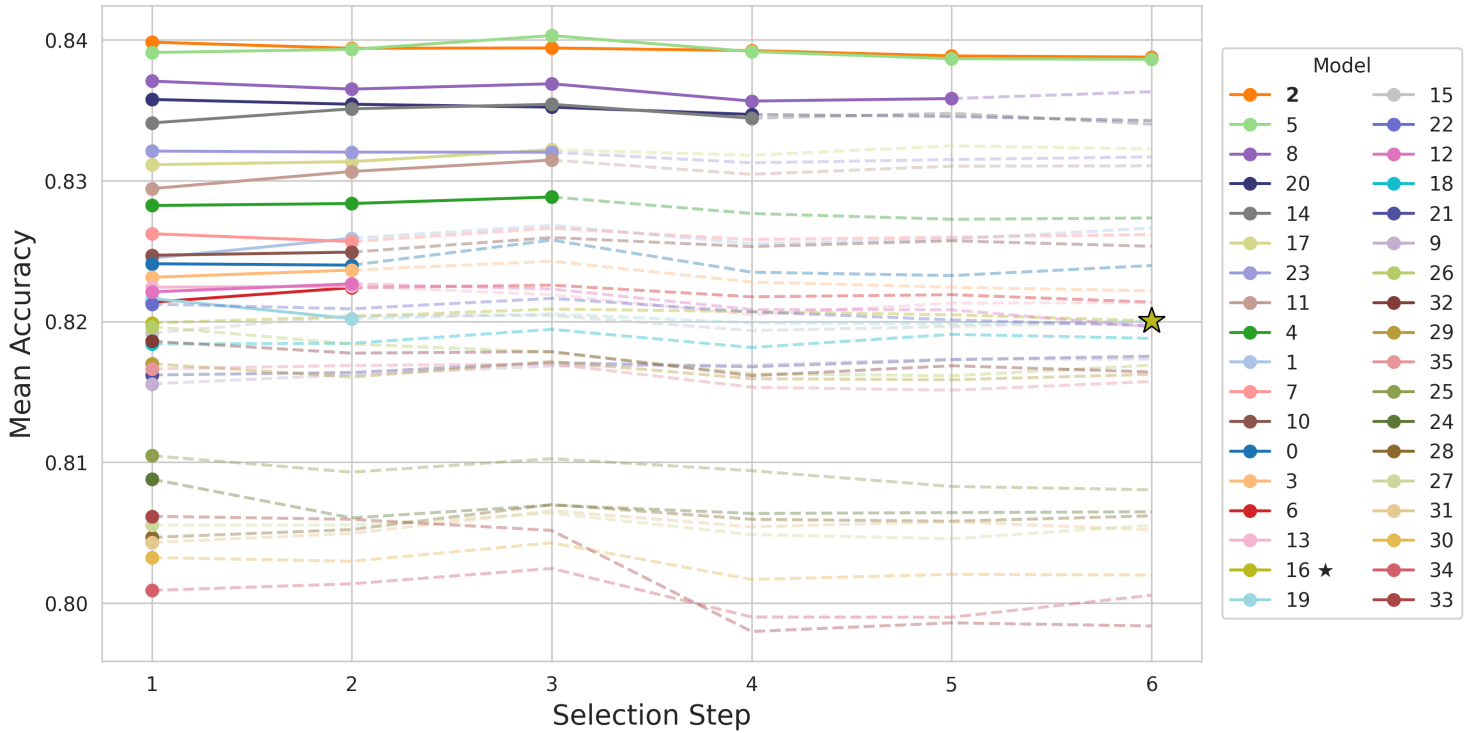
Model	Activation function	Optimizer	Learning rate	Batch size	Regularization
1 (Energy)	ReLU	NAdam	0.0001	64	0.01 (L2)
2 (Energy)		AdamW	0.001	32	0.1 (weight decay)

```
================================================
Layer                Output Shape          Param #
================================================
Model 1 (Energy)  [-1]                     --
|--Conv2d          [-1, 32, 13, 13]        288
|--ReLU            [-1, 32, 13, 13]        --
|--MaxPool2d       [-1, 32, 6, 6]          --
|--Conv2d          [-1, 64, 4, 4]          18,432
|--ReLU            [-1, 64, 4, 4]          --
|--MaxPool2d       [-1, 64, 3, 3]          --
|--Linear          [-1, 9]                 5,193
|--ReLU            [-1, 9]                 --
|--Linear          [-1, 1]                 10
================================================
Trainable params: 23,923
================================================
```



Selected Models: Position Reconstruction

Model Path in the Algorithm: Mean Accuracy



- Train: 10k samples
- ⭐ - base model