A Appendix

A.1 Model Specificites

We used ReLU activation function for our implicit models, transformers have a dropout of 0.1 and a layer norm epsilon of $\epsilon = 10^{-5}$. See Tables 3 and 4 on the next page for model architecture and experiment specificities.

Table 3: Details of training and out-of-distribution test set for each extrapolation task.

Task	**Training Distribution**	**Testing Distribution**
Identity Function	$u_{\text{train}} \in \mathbb{R}^{10,000\times 10} \sim U(-5, 5)$	$u_{\text{test}} \in \mathbb{R}^{3,000\times 10} \sim U(-\kappa, \kappa)$, where κ ranges from 10 to 80
Arithmetic Operations	$u_{\text{train}} \in \mathbb{R}^{10,000\times 50} \sim U(-1, 1)$	$u_{\text{test}} \in \mathbb{R}^{3,000\times 50} \sim U(-\kappa/2, \kappa/2)$, κ ranges from 10 to 10^5
Rolling Average	$u_{\text{train}} \in \mathbb{R}^{10,000\times 10} \sim \mathcal{N}(3, 1)$	$u_{\text{test}} \in \mathbb{R}^{3,000\times 10} \sim \mathcal{N}(3 + \kappa, 1)$, κ ranges from 5 to 100
Rolling Argmax	$u_{\text{train}} \in \mathbb{R}^{10,000\times 10} \sim U(0, 1)$	$u_{\text{test}} \in \mathbb{R}^{3,000\times 10} \sim U(0, \kappa)$, κ ranges from 10^1 to 10^5
Earthquake Location	720,576 (X, Y, Z) locations sampled between (90, -90)°E, 30 features	20,016 samples in each extrapolation region $(90 - 10\kappa, 100 - 10\kappa) \cup (-100 + 10\kappa, -90 + 10k)$, κ ranges from 1 to 9

A.2 Experiments More Results

We provide more in-depth results on the OOD generalization capacities of implicit models for a specific small distribution shift in Figure 12. We compare the training and validation loss on the addition task of both implicit and MLP models where $u_{\text{train}} \in \mathbb{R}^{100} \sim U(1, 2)$ and $u_{\text{val}} \in \mathbb{R}^{100} \sim U(2, 5)$. Even with this small distribution shift, we observe a large improvement.

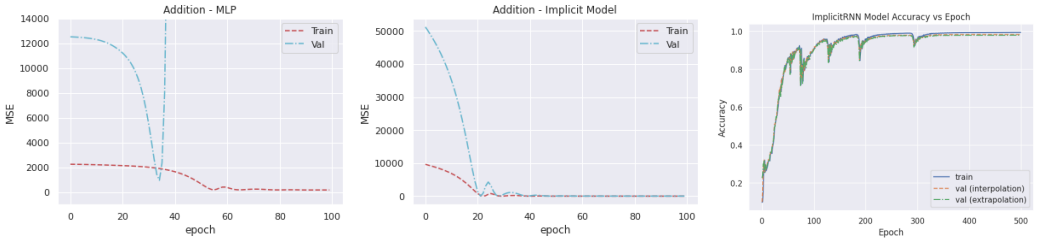

Figure 12: (**left** and **center**) The MLP test loss explodes whereas the implicit model achieves testing loss close to 0. (**right**) For rolling argmax prediction with extrapolation factor t = 10, our implicitRNN performs similarly on interpolated and extrapolated data.

A.3 Data Generation

Spiky Data Generation Both the LSTM and the implicit model were trained on 7000 data points and tested on 3000 data points. The training regime featured 20 spiky regions of 100 data points each. The testing regime featured a proportionate amount of spiky regions. The data points in the spiky regions were sampled from $y = 5 \times (\sin(2x) + \sin(23x) + \sin(78x) + \sin(100x))$. We arbitrarily choose frequencies in $[0, 100]$ to generate a sufficiently spiky pattern. The magnitude of the spiky regions is at most 20. For the non-spiky regimes, the data points were sampled from $y = \sin(x)$ with added noise $\epsilon \sim \mathcal{N}(0, 0.25)$.

Earthquake Data Generation To generate samples of seismic waves between specific longitudes, based on the methods presented by Chuang et al. (2023), we used a 1D velocity model called Ak135 from the Python library obspy.taup. Obspy is a Python framework used to process seismological

Table 4: Details of the explicit and implicit network architectures used in our experiments.

Task	Baseline model	Implicit models	Transformers
Identity Function	MLP: $10 \times 9 \times 9 \times 10$	Regular: $A \in \mathbb{R}^{4\times4}, B \in \mathbb{R}^{4\times10}, C \in \mathbb{R}^{10\times4}, D \in \mathbb{R}^{10\times10}$	Encoder-decoder: $10 \times 10 \times 5$, 5 attention heads
Arithmetic Operations	• MLP: $50\times10\times10\times1$ • NALU: $50 \times 10 \times 10 \times 1$	Regular: $A \in \mathbb{R}^{20\times20}, B \in \mathbb{R}^{20\times50}, C \in \mathbb{R}^{1\times20}, D \in \mathbb{R}^{1\times50}$	• Sequential encoder: 1 layer, 10 attention heads, feedforward dim 50 - processes each array as a single sequence • Depth-wise encoder: 1 layer, 1 attention head, feedforward dim 500, max PE length 50 - processes each element in a given array as a single sequence
Rolling Average	LSTM: $1\times18\times18\times1$	Regular: $A \in \mathbb{R}^{32\times32}, B \in \mathbb{R}^{32\times10}, C \in \mathbb{R}^{10\times32}, D \in \mathbb{R}^{10\times10}$	Encoder-decoder: $10 \times 10 \times 5 \times 10$, 5 attention heads
Rolling Argmax	LSTM: $1\times21\times21\times10$	• Regular: $A \in \mathbb{R}^{36\times36}, B \in \mathbb{R}^{36\times10}, C \in \mathbb{R}^{10\times36}, D \in \mathbb{R}^{10\times10}$ • RNN: $A \in \mathbb{R}^{21\times21}, B \in \mathbb{R}^{21\times23}, C \in \mathbb{R}^{22\times21}, D \in \mathbb{R}^{22\times23}$	• Masked encoder-decoder: 1 encoder layer, 1 decoder layer, 2 attention heads, feedforward dim 10, max PE length 10 • Unmasked encoder-decoder: 1 encoder layer, 1 decoder layer, 2 attention heads, feedforward dim 10, max PE length 10 • Unmasked encoder-decoder without PE: 1 encoder layer, 1 decoder layer, 2 attention heads, feedforward dim 10
Spiky Time Series	LSTM: $1\times20\times20\times1$	RNN: $A \in \mathbb{R}^{20\times20}, B \in \mathbb{R}^{20\times21}, C \in \mathbb{R}^{20\times20}, D \in \mathbb{R}^{20\times21}$ with a 20×1 linear layer	1x10 linear layer (expansion) $\rightarrow$ masked decoder (1 layer, 2 attention heads, feedforward dim 2048, max PE length 10) $\rightarrow$ 10x1 linear layer (contraction)
Volatility Prediction	• LSTM: $1\times38\times38\times1$ • SGD Linear Regression • MLP: $60\times50\times27\times27\times27\times10\times1$	• Regular: $A \in \mathbb{R}^{53\times53}, B \in \mathbb{R}^{53\times60}, C \in \mathbb{R}^{1\times53}, D \in \mathbb{R}^{1\times60}$ • RNN: $A \in \mathbb{R}^{37\times37}, B \in \mathbb{R}^{37\times41}, C \in \mathbb{R}^{40\times37}, D \in \mathbb{R}^{40\times41}$ with a 40×1 linear layer	Sequential encoder (1 layer, 1 attention head, feedforward dim 2048, max PE length 60) $\rightarrow$ 60x1 linear layer
Earthquake Location Prediction	EikoNet: $270 \times 32 \times 128\times128\times128\times32\times4$ (42,500)	Regular: $A \in \mathbb{R}^{190\times190}, B \in \mathbb{R}^{190\times270}, C \in \mathbb{R}^{4\times190}, D \in \mathbb{R}^{4\times270}$ (42,680)	

data. Kennett et al. (1995) demonstrate the accuracy of this model compared to real-world data (see specifically Figure 6). A 1D velocity model assumes the P-wave travel time (the duration the P-wave takes to travel from point A to point B) only depends on two attributes: the distance between the source and the receiver station and the depth of the source. We use this model to create a travel time

lookup table based on these two attributes. We then generate source locations from a mesh that spans the entire globe while adding perturbation to each latitude and longitude pair. We generate station locations using the source-station distances we have from the lookup table and place the stations in random orientations (azimuths) from the source.