

Appendix A Model Details

A.1 Model Sizes

Table 8: All RoMAE model sizes.

Parameter	tiny-shallow	tiny	small	base
d_{model}	180	180	432	720
N_{head}	3	3	6	12
Depth	2	12	12	12
Dim. feed-forward	720	720	1728	2880
Num. parameters	0.782M	4.67M	26.9M	74.7M

We define a set of model sizes for RoMAE which are based on the original BERT [14] and ViT [15] model sizes, and are given in Table 8. The most important difference between RoMAE sizes and the sizes of other BERT-style models is in the d_{model} parameter: RoMAE adopts a different dimensionality because of the constraints regarding Axial RoPE described in Section 3.1. Specifically, we choose d_{model} such that the same model dimensionality works up to 3 positional (axial) dimensions.

Decoder Size: Although we vary the size of the RoMAE Encoder throughout various training runs, we always use RoMAE tiny-shallow as the decoder size when pre-training. Our choice is motivated by MAE [23], which also uses a small and shallow decoder.

A.2 Architectural, Normalization, and Regularization Details

Here we describe the components of RoMAE in more detail, the normalization techniques we use during various training runs and technologies we use to speed up training.

RMSNorm: RoMAE uses RMSNorm [66], which is defined as follows:

$$\bar{a}_i = \frac{a_i}{\text{RMS}(\mathbf{a})} g_i, \quad \text{RMS}(\mathbf{a}) = \sqrt{\epsilon + \frac{1}{d_{\text{model}}} \sum_{i=1}^{d_{\text{model}}} a_i^2} \quad (3)$$

where \mathbf{a} is a sequence of embeddings, g_i is a learned parameter that rescales each $a_i \in \mathbf{a}$, and ϵ is a small value added for numerical stability. Notably, RMSNorm does not centre the input \mathbf{a} like LayerNorm [2] does. Re-centring has been shown not to be necessary and removing it saves compute.

Patch Reconstruction: After passing all [MASK] tokens through the RoMAE decoder, we pass the same reconstruction head over all [MASK] tokens to predict the original patch values. This head consists of an RMSNorm followed by a linear layer $W^{d_{\text{model}} \times n_p}$.

Classification Head: The classification head we use has the same structure as the patch reconstruction head, using an RMSNorm and a linear layer $W^{d_{\text{model}} \times n_{\text{classes}}}$. We place the head on top of the [CLS] token when it is available. Otherwise we take the mean of the output embeddings and place the head on top of this.

Stochastic Depth: In some runs we use stochastic depth [27], which is a form of dropout where whole layers are zeroed out. Specifically, each layer l_m which has depth m , has a probability $\frac{\lambda m}{N_{\text{layers}}}$ to be zeroed out, where λ is the probability of the final layer being zeroed out.

Mixed Precision Training: For some runs we utilize mixed precision training through PyTorch Automatic Mixed Precision (AMP) [4]. When training with AMP, some operations are conducted in a lower precision (either 16-bit brain floating-point (BF16) or 16-bit floating-point (FP16)) instead of the usual 32-bit floating point. This speeds the model up greatly, resulting in significantly less compute resources being used. In our experiments we found that RoMAE still converged well when using mixed precision. We report the precision used in each run along with the hyperparameters.

Dropout: When using dropout [26], we apply it to the attention scores and to the MLP hidden layer with the same probability.

⁴<https://docs.pytorch.org/docs/stable/amp.html>

Label smoothing: To help reduce overfitting, label smoothing [56] prevents the model from becoming overconfident. This is done by changing the model target labels, reducing each correct class label from 1 to a confidence value c , and increasing all incorrect class labels from a value of 0 to a value of $(1 - c)/n_{\text{classes}}$.

Appendix B Additional Experiments and Discussion

B.1 Additional Absolute Position Reconstruction Results

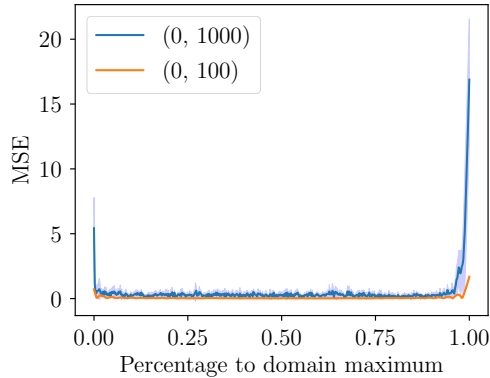


Figure 2: RoMAE position reconstruction MSE across two positional ranges.

Here we provide an additional experiment showing how RoMAE is able to reconstruct absolute position across a wide range of values. To conduct the experiment, we pass only one token into RoMAE, giving it a random position drawn from a uniform distribution $\mathcal{U}_{[a, b]}$, training the model to predict this position. We also pass in the [CLS] token. The experiment is conducted over two domains; (0, 100) and (0, 1000). Hyperparameters and training details are discussed in Section D.5. After training, we evaluate how well the model performs across the range of values it was trained on. Results are plotted in Figure 2.

Discussion: We find that the model is generally able to reconstruct all position values within the two domains tested, except when the position is close to the edges of the domain. This effect occurs already when the domain is relatively small and worsens as it becomes larger. We also find that the model performs better overall on the smaller domain. These results provide empirical support for Proposition 4.2, and show how the model is able to learn to reconstruct absolute position across a large domain. That the loss grows as the position nears the edges of the domain shows that the model does not find solutions that generalize to out-of-distribution positions. These results also indicate that it may be beneficial to rescale positions to be within a smaller range.

B.2 Compute Performance

Table 9: Relative speed of RoMAE when used with regular/irregular positions.

Positional Embedding	Relative speed
Absolute (sin/cos)	1
RoPE (quantized)	0.98
RoPE (continuous)	0.87

We evaluate the performance of RoMAE when using different positional encoding methods, specifically: absolute sin/cos [58], RoPE with integer (quantized) positions, and RoPE with continuous positions. The workload we test on uses 2 positional dimensions for RoPE and 2D image-like inputs to the model. Therefore, this experiment is representative of what one would encounter when using

data such as what we have in the Tiny ImageNet experiment (Section 5.2), or in the ELAsTiCC experiment (Section 5.4). The results, calculated on an NVIDIA 1650Ti GPU, are shown in Table 9.

Although the performance of regular quantized RoPE is not far from standard absolute positional embeddings, when switching to continuous RoPE the model is only 87% of the original speed. This is because we are unable to cache the RoPE frequencies between forward passes. With quantized position on the other hand, everything can be cached once before-hand and reused. While continuous RoPE incurs a notable performance penalty, it is not drastic. We note that other architectures specialized in irregular time-series also suffer from this issue, e.g., ContiFormer [11] is reported as being 6 times slower than the vanilla transformer. The performance of RoMAE could likely be improved through a more optimized RoPE implementation. Quantizing the positions could also address this issue in datasets where it is reasonable to do.

B.3 Extrapolation

Being a BERT-style model, RoMAE is not well suited for extrapolation. During training the bidirectional encoder sees all tokens that lie inside the observed temporal window, therefore it never learns an inductive bias for causal ordering or forward progress in time, and struggles with out-of-distribution positions during inference. Recent work on causal Transformers, for example GPT-family models equipped with RoPE [52] or exponential relative embeddings [54, 53], shows that a strictly unidirectional attention pattern together with position embeddings that extrapolate to unseen indices can capture temporal trends far more effectively. We argue that despite this limitation, RoMAE has a place as a representation learning and interpolation framework, similarly to BERT in language.

B.4 Retaining Frequencies

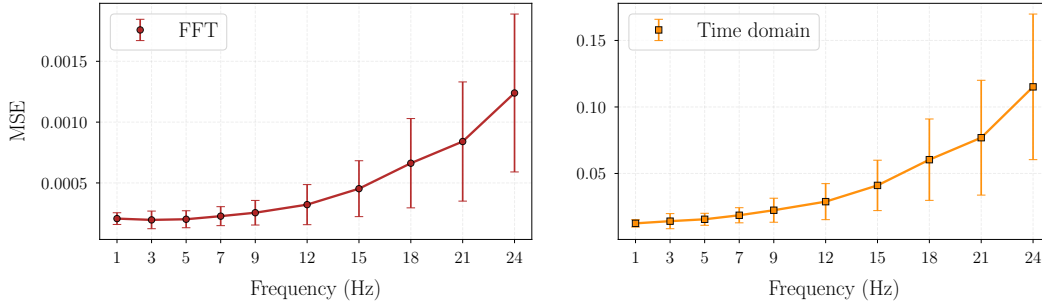


Figure 3: Average MSE obtained from the interpolation task using RoMAE-tiny for time-series with a single varying frequency component. **Left:** MSE computed on the Fast Fourier Transform (FFT). **Right:** MSE in the time domain. We generate 200 time-series per individual frequency, with 50 observed noisy points and 50 masked (interpolated) points, thus a limiting frequency of 25 according to Nyquist-Shannon sampling theorem. Error bars show the standard deviation of the MSE obtained for each individual frequency.

In this appendix we investigate RoMAE’s ability to retain high frequency modes in interpolation tasks. It is known, for example as shown in References [42, 64], that neural networks can exhibit a spectral bias, in that the networks preferentially learn low-frequency components before high-frequency details. Examining how RoMAE reconstructs patterns at different frequencies provides insight into whether the rotational encoding allows to capture fine-grained structure during interpolation, with implications for understanding the inductive biases introduced by this positional encoding scheme.

To empirically assess RoMAE’s ability to reconstruct signals at different frequencies, we designed a controlled toy dataset of noisy sine waves. Each time series is defined over $t \in [0, 1]$ and generated as the sum of one or two frequency components (with equal probability), where integer frequencies are sampled uniformly from $f \in [0, 24]$ Hz and Gaussian noise $\varepsilon \sim \mathcal{N}(0, 0.01)$ is added. Each time-series has 100 data points, 50 of which are taken as input and 50 of which are masked for interpolation. We train RoMAE-tiny for 200 epochs on 10,000 examples. We then evaluate this model on (i) time-series with a single frequency mode as shown in Figure 3 (ii) time-series with two frequencies modes as shown in Figure 4. Using the 50 predicted (interpolated) points, we compute the

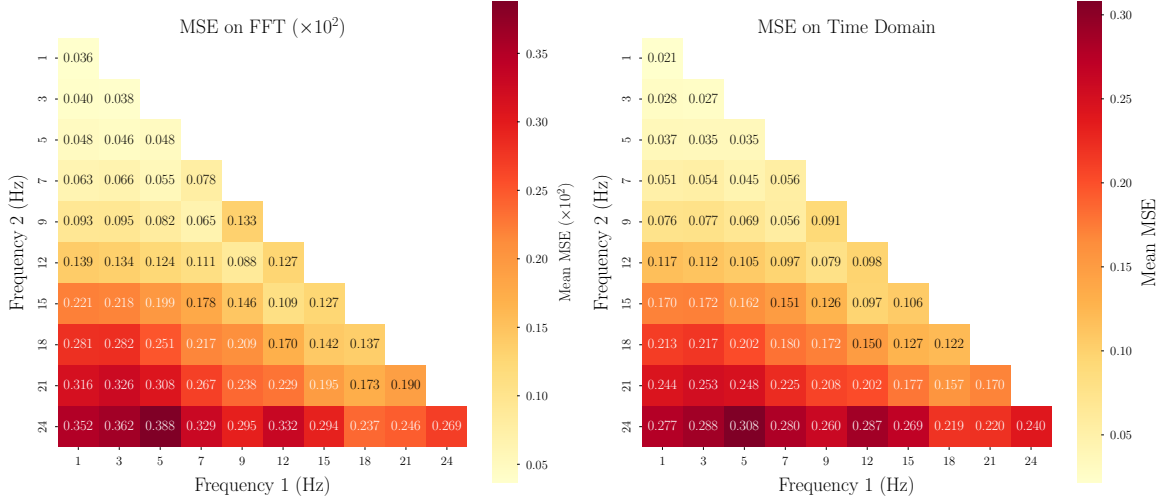


Figure 4: Same as above but now for time-series with two frequency modes present in the signal. **Left:** MSE computed on the FFT. **Right:** MSE in the time domain. The time-series have 50 observed noisy points and 50 masked (interpolated) points.

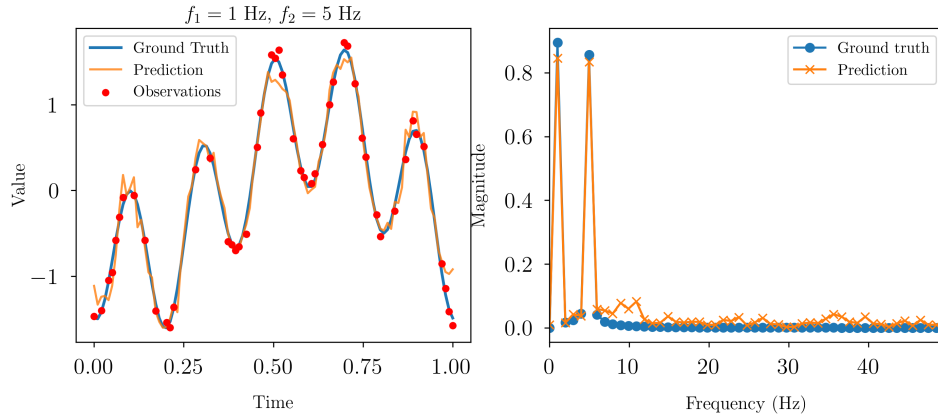


Figure 5: Illustrative realisation from the evaluation of RoMAE on a the bi-frequency time series. **Left:** Interpolation in the time domain for a composite sinusoidal signal with base frequencies 1 and 5 Hz. **Right:** FFT of the ground truth and predicted waveform.

MSE in both the time-domain and Fourier domain. We plot a sample prediction from the evaluation in Figure 5. This analysis was conducted with a mean signal to noise ratio (SNR) of 43.1 over the entire training set. We acknowledge that the distribution of the MSE will be affected by an increasing SNR, resulting in less sensitivity to higher frequency modes.

As expected, we observe a general degradation of the reconstruction for higher frequencies, with an approximately linear trend in error for frequency above 9Hz. For the case of two modes, we observe the same overall trend with a slight preference toward two higher frequency modes, as opposed to one low and one high mode. This is due to the sampling rate of the observations and the fact that the FFT for two higher frequencies has a uni-modal power spectrum, yielding slightly better reconstruction. Lastly, we have checked that the above observations are maintained for non-sinusoidal signals. We repeated the analysis using non-sinusoidal periodic functions, specifically a square wave and a cycloid. We observed similar behaviour for the retention of high frequencies as with the sinusoidal experiment.

Appendix C Proofs

Notation: Here we define additional notation, on top of what is presented in Section 3.1. We define the block-diagonal rotation matrix \mathbf{R} which contains the 2D rotation matrices corresponding to all θ_i 's:

$$\Theta_i = \begin{pmatrix} \cos \theta_i & -\sin \theta_i \\ \sin \theta_i & \cos \theta_i \end{pmatrix}, \quad \mathbf{R} = \begin{pmatrix} \Theta_1 & 0 & \cdots & 0 \\ 0 & \Theta_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \Theta_{d_{\text{model}}/2} \end{pmatrix}. \quad (4)$$

When applying RoPE, \mathbf{R} is exponentiated by position m , then multiplied by x_m . E.g.: $\mathbf{R}^m x_m$.

Definition C.1 ([CLS] token). The learned [CLS] token is a vector $x_{\text{CLS}} \in \mathbb{R}^{d_{\text{model}}}$ consisting of learnable parameters, that is appended to the start of sequence \mathbf{z} as described in Section 4. The position of x_{CLS} is always zero.

C.1 Reconstructing Absolute Position Using the [CLS] Token

We now prove Proposition 4.2 by construction. The proof is based on the proof by Barbero et. al. [4], showing that RoPE can be maximized for any relative distance $r \in \mathbb{Z}$. Here we generalize this result to a continuous position $r \in \mathbb{Q}$.

Proof: Consider a distance $r \in \mathbb{Q}^+ \subset \mathbb{R}$, a query $\mathbf{q} = \psi$ that is non-zero by assumption and a key corresponding to the [CLS] token as described in Definition C.1 such that $\mathbf{k} = \mathbf{R}^r \psi$. Assume that the query is at position $j \in \mathbb{Q}^+$. We compute the dot product between rotated \mathbf{q} and \mathbf{k} :

$$(\mathbf{R}^j \mathbf{q})^\top (\mathbf{R}^0 \mathbf{k}) = \mathbf{q}^\top \mathbf{R}^{-j} \mathbf{k} = \psi^\top \mathbf{R}^{-j+r} \psi \quad (5)$$

We now write this as the sum of dot products between the Θ_i 's and each 2D subspace $\psi^{(i)}$:

$$= \sum_{i=1}^{d_{\text{model}}/2} \left(\psi^{(i)} \right)^\top \Theta_i^{-j+r} \psi^{(i)} \quad (6)$$

$$= \sum_{i=1}^{d_{\text{model}}/2} \left\| \psi^{(i)} \right\|^2 \cos((-j+r)\theta_i) \quad (7)$$

Because both j and r are in \mathbb{Q}^+ , and θ_i is never a multiple of π by definition, the unique maximum occurs when $j = r$. A similar proof applies when $r, j \in \mathbb{Q}^-$. \square

Appendix D Full Experimental Details and Hyperparameters

D.1 Tiny Imagenet Experimental Setup

We present the unified Tiny ImageNet pre-training and fine-tuning hyperparameters in Table 10. All Tiny ImageNet pre-training and fine-tuning runs use the same hyperparameters. Although our final results use FP32, we also tested mixed-precision training and found that FP16 precision works with Tiny-ImageNet as opposed to our experiences on the ELAsTiCC dataset discussed in Section D.6.

When training, we normalize using the ImageNet mean and standard deviation. Each patch is individually normalized when calculating loss as is done in MAE [23]. During fine-tuning, we also use RandAugment [13] with 2 operations and a magnitude of 9. While the model would likely benefit from more epochs during the pre-training and fine-tuning stages, we consider this sufficient for the purposes of an ablation study.

Table 10: Tiny ImageNet unified pre-training and fine-tuning hyperparameters.

Hyperparameter	Pre-Training	Fine-Tuning
Optimizer	AdamW	AdamW
AdamW betas	$\beta_1 = 0.9, \beta_2 = 0.95$	$\beta_1 = 0.9, \beta_2 = 0.999$
Weight Decay	0.05	0.05
Base LR	2×10^{-4}	1×10^{-3}
Batch size	1024	1024
Epochs	200	15
Gradient clip	1	1
Linear LR warmup steps	2000	500
LR schedule	Cosine	Cosine
Dropout	0	0
Stochastic depth λ	0	0
Label smoothing c	–	0.9
Precision	FP32	FP32

D.2 Audio Experimental Setup

In Table 11 we present the unified pre-training and fine-tuning hyperparameters for the audio representation learning and classification experiments, discussed in Section 5.3, in the main text.

When training, both on Audioset [20] and Librispeech [36], we normalize the data using the mean and standard deviation estimated on the entire set of spectrograms. Each patch is individually normalized before calculating the loss as is done in MAE [23].

Table 11: Audio Experiment unified pre-training and fine-tuning hyperparameters.

Hyperparameter	Pre-Training	Fine-Tuning
Optimizer	AdamW	AdamW
AdamW betas	$\beta_1 = 0.9, \beta_2 = 0.95$	$\beta_1 = 0.9, \beta_2 = 0.99$
Weight Decay	0.05	0.02
Base LR	5×10^{-4}	1×10^{-3}
Batch size	64	48
Epochs	150	50
Gradient clip	1	1
Linear LR warmup steps	1000	50
LR schedule	Cosine	Cosine
Dropout	0	0
Stochastic depth λ	0	0
Label smoothing c	–	0.8
Precision	FP32	FP32

D.3 UEA Multivariate Time-series Experimental Setup

We use the same pre-training setting across all datasets, shown in Table 13. When fine-tuning, we find it necessary to have per-dataset hyperparameters. These are shown in Table 12. Because the datasets have varying sizes, resulting in vastly different numbers of training steps, we choose to scale the number of learning rate warmup steps as a percentage of total steps.

D.4 Pendulum Dataset

When training on the Pendulum dataset, we use a custom model size shown in Table 14. The model size is chosen such that the MLP hidden dimension is equal to that of other models benchmarked on the dataset. To create the dataset, we follow the procedure from S5 [51], using their published code⁵.

⁵<https://github.com/lindermanlab/S5/tree/pendulum>

Table 12: UEA Multivariate Time-series Archive fine-tuning hyperparameters. Values that are constant across all runs are reported only once.

Hyperparameter	BM	CT	EP	HB	LSST
Optimizer			SGD		
Momentum			0.9		
Weight Decay			0.		
Base LR	1×10^{-2}	8×10^{-3}	2×10^{-3}	2×10^{-2}	3×10^{-2}
Batch size	8	16	16	32	16
Epochs	50	100	150	30	15
Gradient clip	1	1	1	2	10
Linear LR warmup percentage			10%		
LR schedule			Cosine		
Dropout	0	0	0.2	0	0.2
Stochastic depth λ	0	0	0.2	0	0.2
Label smoothing c	1	0.9	0.8	1	0.9
Precision			FP32		

Table 13: UEA Multivariate Time-series Archive unified pre-training hyperparameters.

Hyperparameter	Value
Optimizer	AdamW
AdamW betas	$\beta_1 = 0.9, \beta_2 = 0.95$
Weight Decay	0.05
Base LR	3×10^{-4}
Batch size	64
Epochs	400
Gradient clip	1
Linear LR warmup percentage	0.1
LR schedule	Cosine
Dropout	0
Stochastic depth λ	0
Precision	FP32

Training RoMAE on the Pendulum dataset is generally very fast because of the small model size and the lack of pre-training.

D.5 Absolute Position Reconstruction Hyperparameters

Here we provide full details for the experiments shown in Section 5.1 and Appendix B.1. These results can be seen in Table 16 and Table 17, respectively. Across both experiments we keep the model size equal to RoMAE-tiny as described in Table 8 except for d_{model} which we set to 960 for the experiment in Section B.1, and set according to the corresponding model size for the experiment in Section 5.1. All experiments in Section 5.1 use the same hyperparameters shown in Table 17. For both experiments we report the mean and standard deviation across 5 different seeds.

D.6 ELAsTiCC Experimental Setup

Full pre-training and fine-tuning hyperparameters for Section 5.4 can be found in Table 18. When creating the train/test split we use the code and pre-processing provided in the ATAT [7] code release⁶. Because the input values in ELAsTiCC are nearly always larger than what one would find with images (e.g., of the order of 10-50 as opposed to between 0 and 1 with images), we found it beneficial to increase the gradient clip threshold to 10 from the common value of 1. In order to handle the variable number of points per sample we utilize padding, applying a pad mask to the attention scores. Although our final model was trained using full FP32 precision, we tested RoMAE with both FP16

⁶<https://github.com/alercbroker/ATAT>

Table 14: Pendulum dataset custom model size.

Model Parameter	Value
d_{model}	60
N_{head}	2
Depth	2
Dim. feed-forward	30
Num. parameters	37.4K

Table 15: Pendulum dataset end-to-end training hyperparameters.

Hyperparameter	Value
Optimizer	AdamW
AdamW betas	$\beta_1 = 0.9, \beta_2 = 0.999$
Weight Decay	0.01
Base LR	3×10^{-4}
Batch size	16
Epochs	50
Gradient clip	1
Linear LR warmup steps	1000
LR schedule	Cosine
Dropout	0
Stochastic depth λ	0
Precision	FP32

Table 16: End-to-end training hyperparameters for the absolute reconstruction range experiment (Section B.1). Values that are constant across all runs are reported only once.

Hyperparameter	(0, 100)	(0, 1000)
Optimizer	SGD	
Momentum	0.9	
Weight Decay	0.	
Base LR	5×10^{-6}	5×10^{-7}
Batch size	64	
Epochs	10	
Gradient clip	inf	
Linear LR warmup steps	625	
LR schedule	Cosine	
Dropout	0	
Stochastic depth λ	0	
Precision	FP32	

Table 17: End-to-end training hyperparameters for the absolute reconstruction MSE experiment (Section 5.1).

Hyperparameter	Value
Optimizer	AdamW
AdamW betas	$\beta_1 = 0.9, \beta_2 = 0.999$
Weight Decay	0.01
Base LR	5×10^{-4}
Batch size	64
Epochs	10
Gradient clip	inf
Linear LR warmup steps	625
LR schedule	Cosine
Dropout	0
Stochastic depth λ	0
Precision	FP32

and BF16 for mixed precision training, and found that FP16 resulted in NaN values. This is likely due to the increased input range in ELAsTiCC interacting poorly with the reduced range of FP16. We found that BF16, with its larger range, worked well.

Each light curve in the ELAsTiCC dataset contains recordings of both the flux difference and variance. An example training sample is visualized in Figure 6, showing how each band has a different number

Table 18: ELAsTiCC full training hyperparameters.

Hyperparameter	Pre-Training	Fine-Tuning
Optimizer	AdamW	AdamW
AdamW betas	$\beta_1 = 0.9, \beta_2 = 0.95$	$\beta_1 = 0.9, \beta_2 = 0.999$
Weight Decay	0.05	0.05
Base LR	6.4×10^{-3}	8×10^{-4}
Batch size	16384	4096
Epochs	200	25
Gradient clip	10	10
Linear LR warmup steps	2000	2000
LR schedule	Cosine	Cosine
Dropout	0	0.2
Stochastic depth λ	0	0.2
Precision	FP32	FP32

of observations, each of which is at a different time than the others. To convert each point in the light curve to an embedding we use a patch size of (1, 2) for time and flux/variance respectively. Therefore, during pre-training the model predicts not only the masked flux difference values but also variance for each point, while time and band index are embedded using position.

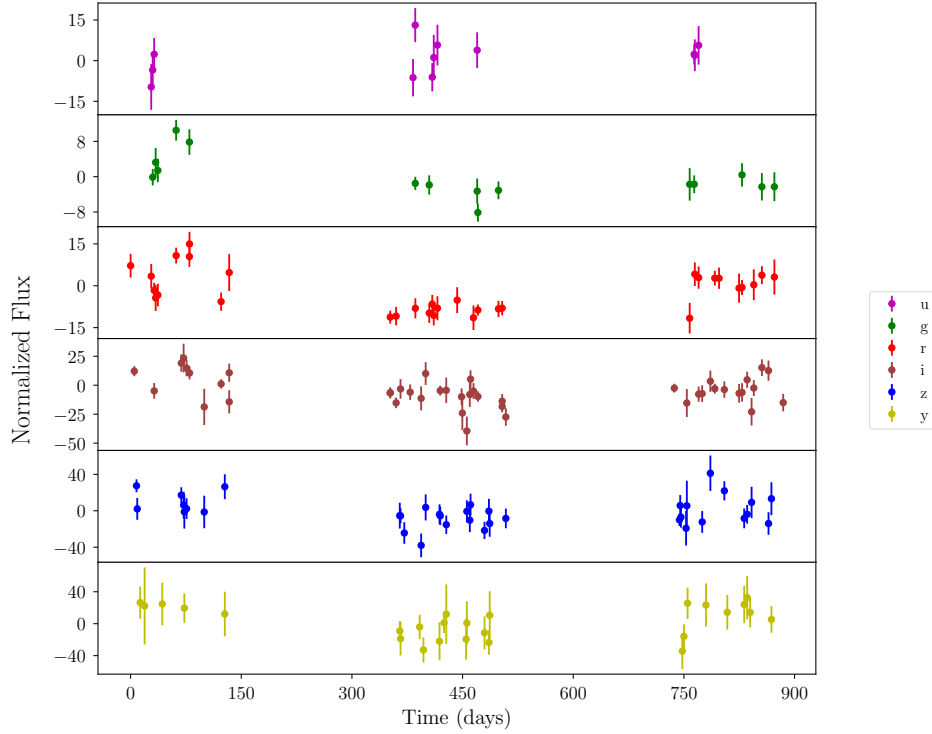


Figure 6: A training example from the ELAsTiCC dataset. The flux difference of each band has already been normalized.

D.7 Spiral dataset

We construct a dataset of 300 spirals as per the prescription from Ref. [12], similarly allocating 200 for training and 100 for testing. Each spiral is randomly assigned to be either clockwise or counter-clockwise, with parameters drawn from normal distributions

$$a \sim \mathcal{N}(0, \alpha) \quad \text{and} \quad b \sim \mathcal{N}(0.3, \alpha),$$

Table 19: End-to-end training hyperparameters for the spirals dataset.

Hyperparameter	Value
Model size	Tiny
Optimizer	AdamW
AdamW betas	$\beta_1 = 0.9, \beta_2 = 0.999$
Weight Decay	0.01
Base LR	3×10^{-4}
Batch size	32
Epochs	500
Gradient clip	1
Linear LR warmup steps	2000
LR schedule	Cosine
Dropout	0
Stochastic depth λ	0
Precision	FP32

where $\alpha = 0.02$. For the results presented in Table 7 and for comparison with ContiFormer, we add Gaussian noise sampled from $\mathcal{N}(0, \beta)$ to the training samples, setting $\beta = 0.1$. The spirals were truncated at times corresponding to 6π in both cases, and only the parts of the spiral corresponding to the interpolation task carried out by Ref. [12] were used. Each spiral is discretized into 75 evenly spaced time steps. To create irregular time series data, 30 time points are randomly selected from the first half of each spiral, which are used for interpolation. We show the model hyperparameters used to generate the results in Table 19.

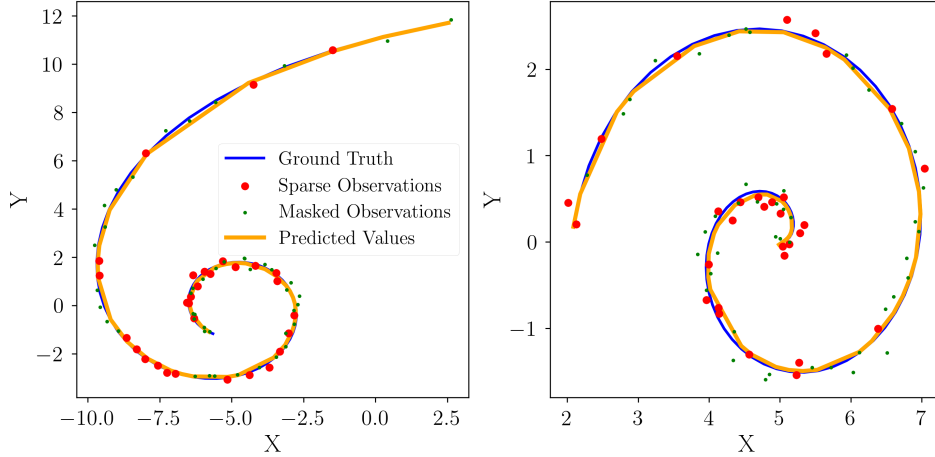


Figure 7: Two sample realisations of differing chirality from the test set of spirals. The green line is the ground truth trajectory. The Red points are the 75 stochastic inputs of which 45 are masked. The blue points are the interpolated predictions.

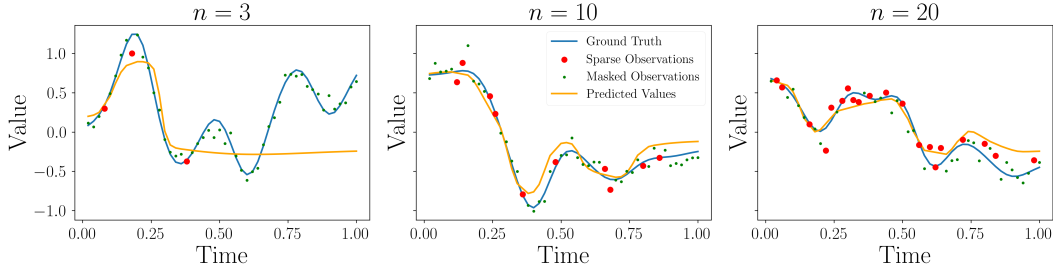
The x and y coordinates of the spirals are embedded using a patch size of (1,2) for time and x/y coordinates respectively. We train for 400 epochs with a learning rate of 10^{-3} . Uncertainties presented in Table 7 represent the evaluation uncertainties after 10 trials with randomly seeded batches of 100 test spirals. The addition of the CLS token was observed to not significantly improve performance. The code for generating the exact spirals used for this experiment, as well the details of the experimental setup for ContiFormer on their github [7].

Table 20: Synthetic dataset training hyperparameters

Hyperparameter	Value
Model size	Tiny
Optimizer	AdamW
AdamW betas	$\beta_1 = 0.9, \beta_2 = 0.999$
Weight Decay	0.01
Base LR	1×10^{-3}
Batch size	8
Epochs	50
Gradient clip	1
Linear LR warmup steps	200
LR schedule	Cosine
Dropout	0
Stochastic depth λ	0
Precision	FP32

D.8 Synthetic dataset

We evaluate RoMAE on a synthetic interpolation task introduced by Ref. [47], using the same code to generate the dataset [8]. The dataset comprises 2,000 univariate time series, each with 50 uniformly spaced time points in $[0, 1]$. With a patch size of $(1, 1)$, each individual point is converted to a token. Each trajectory is generated by sampling 10 latent variables $z_k \sim \mathcal{N}(0, 1)$ at reference times $r_k = 0.1 \cdot k$, and applying an RBF kernel smoother with bandwidth $\alpha = 120.0$ to interpolate values across the timeline. Gaussian noise $\mathcal{N}(0, 0.01)$ is added to simulate measurement error. To mimic irregular sampling, we randomly select between 3 and 10 observed points per trajectory. The dataset is split into 80% training, 10% validation, and 10% test sets. Performance is assessed using mean squared error (MSE). We display results from some test samples for a number of observed points $n = 2, 10$ and 20 in Figure 8. The addition of the CLS token was observed to not significantly impact performance. We show the model hyperparameters used to generate the results in Table 20.

Figure 8: Samples from interpolation tests using $n = 3, 10$ and 20 observations.

D.9 PhysioNet

We adopt the pre-processed release of the PHYSIONET/CinC 2012 Challenge [50], comprising multivariate clinical time-series collected during the 48h window following intensive-care-unit (ICU) admission. Static covariates (*Age, Gender, Height, ICU type*) occupy feature indices 0–3 and are always observed, whereas the remaining 37 channels are sparsely and irregularly sampled.

In order to benchmark RoMAE we directly compare performance on the interpolation task using the same pre-processed version of the dataset produced by Ref. [35]⁹, which rounds the observation times to the nearest minute resulting in 2880 possible measurement times per time series. The

⁸<https://github.com/microsoft/SeqML/tree/main/ContiFormer>
⁹<https://github.com/reml-lab/hetvae/blob/main/src/utils.py>
⁹<https://github.com/reml-lab/hetvae>

data set includes 8000 instances that can be used for interpolation experiments. We additionally use the same experimental protocols which involve masking 50% of observed time points. Each multivariate record in the PHYSIONET 48 h clinical dataset is converted into a sequence of *scalar tokens* that RoMAE can process. Let $x_t^{(d)} \in \mathbb{R}$ denote the value of feature $d \in \{1, \dots, 41\}$ measured at minute-resolution time step $t \in \{1, \dots, T\}$ ($T \leq 2880$); let $m_t^{(d)} \in \{0, 1\}$ be the corresponding observation mask (1 = measured). We flatten the spatio-temporal grid into a one-dimensional token list $\{(x_n, p_n)\}_{n=1}^N$ with

$$x_n = x_t^{(d)}, \quad p_n = [t/T, d]^\top, \quad N = \sum_{t,d} 1.$$

The two-dimensional positional vector p_n encodes (i) the *normalised time* $t/T \in [0, 1]$ and (ii) the *feature index* d , providing the $n_{\text{pos}} = 2$ co-ordinates required by RoMAE. During training we stochastically subsample 50% of the observed tokens. The final input tensor hence has length N for the values, and shape $(2, N)$ for the positions, and a Boolean mask of length N indicating which tokens RoMAE must reconstruct, exactly matching the interpolation protocol of the HeTVAE benchmark.

We show the results of interpolation study in Table 7 where we compare to HetVAE [35], as well as 8 other models benchmarked in that study. We show the model hyperparameters used to generate the results in Table 21 along with the addition of the CLS token that was seen to improve the results. Lastly, official Physionet challenge can be found on their website¹⁰

Table 21: PhysioNet dataset training hyperparameters

Hyperparameter	Value
Model size	Tiny
Optimizer	AdamW
AdamW betas	$\beta_1 = 0.9, \beta_2 = 0.999$
Weight Decay	0.01
Base LR	1×10^{-4}
Batch size	16
Epochs	200
Gradient clip	1
Linear LR warmup steps	100
LR schedule	Cosine
Dropout	0
Stochastic depth λ	0
Precision	FP32

¹⁰<https://physionet.org/content/challenge-2012/1.0.0/>

References

- [1] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lucic, and C. Schmid. Vivit: A video vision transformer. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 6816–6826. IEEE, 2021. doi: 10.1109/ICCV48922.2021.00676. URL <https://doi.org/10.1109/ICCV48922.2021.00676>.
- [2] L. J. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016. URL <http://arxiv.org/abs/1607.06450>.
- [3] A. J. Bagnall, H. A. Dau, J. Lines, M. Flynn, J. Large, A. Bostrom, P. Southam, and E. J. Keogh. The UEA multivariate time series classification archive, 2018. *CoRR*, abs/1811.00075, 2018. URL <http://arxiv.org/abs/1811.00075>.
- [4] F. Barbero, A. Vitvitskiy, C. Perivolaropoulos, R. Pascanu, and P. Velickovic. Round and round we go! what makes rotary positional encodings useful? In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL <https://openreview.net/forum?id=GtvuNrK58a>.
- [5] P. Becker, H. Pandya, G. H. W. Gebhardt, C. Zhao, C. J. Taylor, and G. Neumann. Recurrent kalman networks: Factorized inference in high-dimensional deep feature spaces. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 544–552. PMLR, 2019. URL <http://proceedings.mlr.press/v97/becker19a.html>.
- [6] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL <https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfbcb4967418bfb8ac142f64a-Abstract.html>.
- [7] Cabrera-Vives, G., Moreno-Cartagena, D., Astorga, N., Reyes-Jainaga, I., Förster, F., Huijse, P., Arredondo, J., Muñoz Arancibia, A. M., Bayo, A., Catelan, M., Estévez, P. A., Sánchez-Sáez, P., Álvarez, A., Castellanos, P., Gallardo, P., Moya, A., and Rodríguez-Mancini, D. Atat: Astronomical transformer for time series and tabular data. *A&A*, 689:A289, 2024. doi: 10.1051/0004-6361/202449475. URL <https://doi.org/10.1051/0004-6361/202449475>.
- [8] H. Chen, Y. Han, F. Chen, X. Li, Y. Wang, J. Wang, Z. Wang, Z. Liu, D. Zou, and B. Raj. Masked autoencoders are effective tokenizers for diffusion models. *CoRR*, abs/2502.03444, 2025. doi: 10.48550/ARXIV.2502.03444. URL <https://doi.org/10.48550/arXiv.2502.03444>.
- [9] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud. Neural Ordinary Differential Equations. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 6571–6583, 2018.
- [10] T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud. Neural ordinary differential equations. In S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 6572–6583, 2018. URL <https://proceedings.neurips.cc/paper/2018/hash/69386f6bb1dfed68692a24c8686939b9-Abstract.html>.
- [11] Y. Chen, K. Ren, Y. Wang, Y. Fang, W. Sun, and D. Li. Contiformer: Continuous-time transformer for irregular time series modeling. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 -*

- 16, 2023, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/9328208f88ec69420031647e6ff97727-Abstract-Conference.html.
- [12] Y. Chen, Q. Wang, and Y. e. Fu. Continuous-time Transformer for Irregular Time-series Predictions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.
- [13] E. D. Cubuk, B. Zoph, J. Shlens, and Q. Le. Randaugment: Practical automated data augmentation with a reduced search space. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL <https://proceedings.neurips.cc/paper/2020/hash/d85b63ef0ccb114d0a3bb7b7d808028f-Abstract.html>.
- [14] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, and T. Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/V1/N19-1423. URL <https://doi.org/10.18653/v1/n19-1423>.
- [15] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL <https://openreview.net/forum?id=YicbFdNTTy>.
- [16] S. Elfving, E. Uchibe, and K. Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 107:3–11, 2018. doi: 10.1016/J.NEUNET.2017.12.012. URL <https://doi.org/10.1016/j.neunet.2017.12.012>.
- [17] Y. Fang, Q. Sun, X. Wang, T. Huang, X. Wang, and Y. Cao. EVA-02: A visual representation for neon genesis. *Image Vis. Comput.*, 149:105171, 2024. doi: 10.1016/J.IMAVIS.2024.105171. URL <https://doi.org/10.1016/j.imavis.2024.105171>.
- [18] C. Feichtenhofer, H. Fan, Y. Li, and K. He. Masked autoencoders as spatiotemporal learners. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/e97d1081481a4017df96b51be31001d3-Abstract-Conference.html.
- [19] S. Gao, T. Koker, O. Queen, T. Hartvigsen, T. Tsiligkaridis, and M. Zitnik. Units: A unified multi-task time series model. *Advances in Neural Information Processing Systems*, 37:140589–140631, 2024.
- [20] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 776–780. IEEE, 2017.
- [21] Y. Gong, C. Lai, Y. Chung, and J. R. Glass. SSAST: self-supervised audio spectrogram transformer. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 10699–10709. AAAI Press, 2022. doi: 10.1609/AAAI.V36I10.21315. URL <https://doi.org/10.1609/aaai.v36i10.21315>.
- [22] A. Gu and T. Dao. Mamba: Linear-time sequence modeling with selective state spaces. *CoRR*, abs/2312.00752, 2023. doi: 10.48550/ARXIV.2312.00752. URL <https://doi.org/10.48550/arXiv.2312.00752>.

- [23] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. B. Girshick. Masked autoencoders are scalable vision learners. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 15979–15988. IEEE, 2022. doi: 10.1109/CVPR52688.2022.01553. URL <https://doi.org/10.1109/CVPR52688.2022.01553>.
- [24] D. Hendrycks and K. Gimpel. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *CoRR*, abs/1606.08415, 2016. URL <http://arxiv.org/abs/1606.08415>.
- [25] B. Heo, S. Park, D. Han, and S. Yun. Rotary position embedding for vision transformer. In A. Leonardis, E. Ricci, S. Roth, O. Russakovsky, T. Sattler, and G. Varol, editors, *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part X*, volume 15068 of *Lecture Notes in Computer Science*, pages 289–305. Springer, 2024. doi: 10.1007/978-3-031-72684-2_17. URL https://doi.org/10.1007/978-3-031-72684-2_17.
- [26] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580, 2012. URL <http://arxiv.org/abs/1207.0580>.
- [27] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger. Deep networks with stochastic depth. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV*, volume 9908 of *Lecture Notes in Computer Science*, pages 646–661. Springer, 2016. doi: 10.1007/978-3-319-46493-0_39. URL https://doi.org/10.1007/978-3-319-46493-0_39.
- [28] P. Huang, H. Xu, J. Li, A. Baevski, M. Auli, W. Galuba, F. Metze, and C. Feichtenhofer. Masked autoencoders that listen. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*. URL http://papers.nips.cc/paper_files/paper/2022/hash/b89d5e209990b19e33b418e14f323998-Abstract-Conference.html.
- [29] P. Jeevan and A. Sethi. Resource-efficient hybrid x-formers for vision. In *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2022, Waikoloa, HI, USA, January 3-8, 2022*, pages 3555–3563. IEEE, 2022. doi: 10.1109/WACV51458.2022.00361. URL <https://doi.org/10.1109/WACV51458.2022.00361>.
- [30] P. Kidger, J. Morrill, J. Foster, and T. J. Lyons. Neural controlled differential equations for irregular time series. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL <https://proceedings.neurips.cc/paper/2020/hash/4a5876b450b45371f6cfe5047ac8cd45-Abstract.html>.
- [31] Y. Le and X. S. Yang. Tiny imagenet visual recognition challenge. 2015. URL http://vision.stanford.edu/teaching/cs231n/reports/2015/pdfs/yle_project.pdf.
- [32] Z. Li, S. Li, and X. Yan. Time series as images: Vision transformer for irregularly sampled time series. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*. URL http://papers.nips.cc/paper_files/paper/2023/hash/9a17c1eb808cf012065e9db47b7ca80d-Abstract-Conference.html.
- [33] Y. Liu, H. Zhang, C. Li, X. Huang, J. Wang, and M. Long. Timer: Generative pre-trained transformers are large time series models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL <https://openreview.net/forum?id=bYRYb7DMNo>.
- [34] Z. Lu, Z. Wang, D. Huang, C. Wu, X. Liu, W. Ouyang, and L. Bai. Fit: Flexible vision transformer for diffusion model. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL <https://openreview.net/forum?id=jZVen2JguY>.

- [35] M. Moor, B. Rieck, M. Horn, C. R. Jutzeler, and K. Borgwardt. Early Recognition of Sepsis with Heteroscedastic Temporal Variational Autoencoders. In *International Conference on Machine Learning (ICML)*, pages 7781–7792, 2021.
- [36] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur. Librispeech: An asr corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210, 2015. doi: 10.1109/ICASSP.2015.7178964.
- [37] H. Patel, R. Qiu, A. Irwin, S. Sadiq, and S. Wang. EMIT - event-based masked auto encoding for irregular time series. In E. Baralis, K. Zhang, E. Damiani, M. Debbah, P. Kalnis, and X. Wu, editors, *IEEE International Conference on Data Mining, ICDM 2024, Abu Dhabi, United Arab Emirates, December 9-12, 2024*, pages 370–379. IEEE, 2024. doi: 10.1109/ICDM59182.2024.00044. URL <https://doi.org/10.1109/ICDM59182.2024.00044>.
- [38] B. Peng, J. Quesnelle, H. Fan, and E. Shippole. Yarn: Efficient context window extension of large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL <https://openreview.net/forum?id=wHBfxhZu1u>.
- [39] K. J. Piczak. Esc: Dataset for environmental sound classification. In *Proceedings of the 23rd ACM International Conference on Multimedia, MM '15*, page 1015–1018, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450334594. doi: 10.1145/2733373.2806390. URL <https://doi.org/10.1145/2733373.2806390>.
- [40] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al. Improving language understanding by generative pre-training. *OpenAI blog*, 2018. URL https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.
- [41] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 2019. URL https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.
- [42] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. Hamprecht, Y. Bengio, and A. Courville. On the spectral bias of neural networks. In *International conference on machine learning*, pages 5301–5310. PMLR, 2019.
- [43] B. Rozière, J. Gehring, F. Gloeckle, S. Sootla, I. Gat, X. E. Tan, Y. Adi, J. Liu, T. Remez, J. Rapin, A. Kozhevnikov, I. Evtimov, J. Bitton, M. Bhatt, C. Canton-Ferrer, A. Grattafiori, W. Xiong, A. Défossez, J. Copet, F. Azhar, H. Touvron, L. Martin, N. Usunier, T. Scialom, and G. Synnaeve. Code llama: Open foundation models for code. *CoRR*, abs/2308.12950, 2023. doi: 10.48550/ARXIV.2308.12950. URL <https://doi.org/10.48550/arXiv.2308.12950>.
- [44] Y. Rubanova, T. Q. Chen, and D. Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. B. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5321–5331, 2019. URL <https://proceedings.neurips.cc/paper/2019/hash/42a6845a557bef704ad8ac9cb4461d43-Abstract.html>.
- [45] M. Schirmer, M. Eltayeb, S. Lessmann, and M. Rudolph. Modeling irregular time series with continuous recurrent units. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvári, G. Niu, and S. Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 19388–19405. PMLR, 2022. URL <https://proceedings.mlr.press/v162/schirmer22a.html>.
- [46] S. N. Shukla and B. M. Marlin. Multi-time attention networks for irregularly sampled time series. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=4c0J61wQ4_.

- [47] S. N. Shukla and B. M. Marlin. Heteroscedastic temporal variational autoencoder for irregularly sampled time series. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL <https://openreview.net/forum?id=Az7opqbQE-3>.
- [48] S. N. Shukla and B. M. Marlin. Heteroscedastic temporal variational autoencoder for irregularly sampled time series. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL <https://openreview.net/forum?id=Az7opqbQE-3>.
- [49] I. Silva, B. Moody, D. Scott, L. Celi, R. Mark, and G. Clifford. The PhysioNet/Computing in Cardiology Challenge 2012: Predicting In-Hospital Mortality from ICU Data. In *Computing in Cardiology*, pages 245–248, 2012.
- [50] I. Silva, G. Moody, D. J. Scott, L. A. Celi, and R. G. Mark. Predicting in-hospital mortality of icu patients: The physionet/computing in cardiology challenge 2012. In *2012 computing in cardiology*, pages 245–248. IEEE, 2012.
- [51] J. T. H. Smith, A. Warrington, and S. W. Linderman. Simplified state space layers for sequence modeling. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL <https://openreview.net/forum?id=Ai8Hw3AXqks>.
- [52] Z. Song, Q. Lu, H. Zhu, D. Buckeridge, and Y. Li. Trajgpt: Irregular time-series representation learning for health trajectory analysis. *CoRR*, abs/2410.02133, 2024. doi: 10.48550/ARXIV.2410.02133. URL <https://doi.org/10.48550/arXiv.2410.02133>.
- [53] N. Stroh. Trackgpt—a generative pre-trained transformer for cross-domain entity trajectory forecasting. *arXiv preprint arXiv:2402.00066*, 2024.
- [54] J. Su, M. H. M. Ahmed, Y. Lu, S. Pan, W. Bo, and Y. Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. doi: 10.1016/J.NEUCOM.2023.127063. URL <https://doi.org/10.1016/j.neucom.2023.127063>.
- [55] K. Su, Q. Wu, P. Cai, X. Zhu, X. Lu, Z. Wang, and K. Hu. RI-MAE: rotation-invariant masked autoencoders for self-supervised point cloud representation learning. In T. Walsh, J. Shah, and Z. Kolter, editors, *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pages 7015–7023. AAAI Press, 2025. doi: 10.1609/AAAI.V39I7.32753. URL <https://doi.org/10.1609/aaai.v39i7.32753>.
- [56] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2818–2826. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.308. URL <https://doi.org/10.1109/CVPR.2016.308>.
- [57] Z. Tong, Y. Song, J. Wang, and L. Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/416f9cb3276121c42eebb86352a4354a-Abstract-Conference.html.
- [58] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017. URL <https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html>.

- [59] L. Wang, B. Huang, Z. Zhao, Z. Tong, Y. He, Y. Wang, Y. Wang, and Y. Qiao. Videomae V2: scaling video masked autoencoders with dual masking. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 14549–14560. IEEE, 2023. doi: 10.1109/CVPR52729.2023.01398. URL <https://doi.org/10.1109/CVPR52729.2023.01398>.
- [60] Q. Wen, T. Zhou, C. Zhang, W. Chen, Z. Ma, J. Yan, and L. Sun. Transformers in time series: A survey. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China*, pages 6778–6786. ijcai.org, 2023. doi: 10.24963/IJCAI.2023/759. URL <https://doi.org/10.24963/ijcai.2023/759>.
- [61] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45, 2020.
- [62] W. Xiong, J. Liu, I. Molybog, H. Zhang, P. Bhargava, R. Hou, L. Martin, R. Rungta, K. A. Sankararaman, B. Oguz, M. Khabsa, H. Fang, Y. Mehdad, S. Narang, K. Malik, A. Fan, S. Bhosale, S. Edunov, M. Lewis, S. Wang, and H. Ma. Effective long-context scaling of foundation models. In K. Duh, H. Gómez-Adorno, and S. Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 4643–4663. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.NAACL-LONG.260. URL <https://doi.org/10.18653/v1/2024.naacl-long.260>.
- [63] M. Xu, X. Men, B. Wang, Q. Zhang, H. Lin, X. Han, and W. Chen. Base of rope bounds context length. In A. Globersons, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. M. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/9f12dd32d552f3ad9eaa0e9dfec291be-Abstract-Conference.html.
- [64] Z.-Q. J. Xu, Y. Zhang, and T. Luo. Overview frequency principle/spectral bias in deep learning. *Communications on Applied Mathematics and Computation*, 7(3):827–864, 2025.
- [65] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, and C. Eickhoff. A transformer-based framework for multivariate time series representation learning. In F. Zhu, B. C. Ooi, and C. Miao, editors, *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*, pages 2114–2124. ACM, 2021. doi: 10.1145/3447548.3467401. URL <https://doi.org/10.1145/3447548.3467401>.
- [66] B. Zhang and R. Sennrich. Root mean square layer normalization. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. B. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 12360–12371, 2019. URL <https://proceedings.neurips.cc/paper/2019/hash/1e8a19426224ca89e83cef47f1e7f53b-Abstract.html>.