Zero-Shot Forecasting and Neural Operators
Master Lab

Arwin — [*1] **Jan Hardtke** [*12] **Nadia Gharbi** [2]

Abstract

A

1. Structure

- abstract
- a brief introduction (Nadia)
- a discussion of related work (Decoder Only, Statistical models like ARIMA, (briefly FIM-l and DeepONet with reference to chapter conceptual aspects) ... do more research...) (Nadia)
- An explanation of the main conceptual aspects (e.g. DeepONets, random function generation and maybe zero-shot learning) is expected. (Arwin)
- In an experimental section, give details for reproducibility of your work (e.g. hyperparameters, training objectives, hardware, runtime...). (Jan)
- A results section could include a description of the tasks and your findings (use tables and figures to present your results).
- Finish the report with a short concluding section.

- Note that you are encouraged to further improve and explore the models for each task! (Jan)

2. Electronic Submission

Submission to ICML 2025 will be entirely electronic, via a web site (not email). Information about the submission process and LaTeX templates are available on the conference web site at:

`http://icml.cc/`

The guidelines below will be enforced for initial submissions and camera-ready copies. Here is a brief summary:

[*]Equal contribution [1]Department of XXX, University of YYY, Location, Country [2]Company Name, Location, Country. Correspondence to: Firstname1 Lastname1 <first1.last1@xxx.edu>, Firstname2 Lastname2 <first2.last2@www.uk>.

Proceedings of the 41st International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

- Submissions must be in PDF.
- If your paper has appendices, submit the appendix together with the main body and the references **as a single file**. Reviewers will not look for appendices as a separate PDF file. So if you submit such an extra file, reviewers will very likely miss it.
- Page limit: The main body of the paper has to be fitted to 8 pages, excluding references and appendices; the space for the latter two is not limited in pages, but the total file size may not exceed 10MB. For the final version of the paper, authors can add one extra page to the main body.
- **Do not include author information or acknowledgements** in your initial submission.
- Your paper should be in **10 point Times font**.
- Make sure your PDF file only uses Type-1 fonts.
- Place figure captions *under* the figure (and omit titles from inside the graphic file itself). Place table captions *over* the table.
- References must include page numbers whenever possible and be as complete as possible. Place multiple citations in chronological order.
- Do not alter the style template; in particular, do not compress the paper format by reducing the vertical spaces.
- Keep your abstract brief and self-contained, one paragraph and roughly 4–6 sentences. Gross violations will require correction at the camera-ready phase. The title should have content words capitalized.

2.1. Submitting Papers

Anonymous Submission: ICML uses double-blind review: no identifying author information may appear on the title page or in the paper itself. Section 3.3 gives further details.

Authors must provide their manuscripts in **PDF** format. Furthermore, please make sure that files contain only embedded

Type-1 fonts (e.g., using the program `pdffonts` in linux or using File/DocumentProperties/Fonts in Acrobat). Other fonts (like Type-3) might come from graphics files imported into the document.

Authors using **Word** must convert their document to PDF. Most of the latest versions of Word have the facility to do this automatically. Submissions will not be accepted in Word format or any format other than PDF. Really. We're not joking. Don't send Word.

Those who use **LaTeX** should avoid including Type-3 fonts. Those using `latex` and `dvips` may need the following two commands:

```
dvips -Ppdf -tletter -G0 -o paper.ps paper.dvi
ps2pdf paper.ps
```

It is a zero following the "-G", which tells dvips to use the config.pdf file. Newer TEX distributions don't always need this option.

Using `pdflatex` rather than `latex`, often gives better results. This program avoids the Type-3 font problem, and supports more advanced features in the `microtype` package.

Graphics files should be a reasonable size, and included from an appropriate format. Use vector formats (.eps/.pdf) for plots, lossless bitmap formats (.png) for raster graphics with sharp lines, and jpeg for photo-like images.

The style file uses the `hyperref` package to make clickable links in documents. If this causes problems for you, add `nohyperref` as one of the options to the `icml2025` usepackage statement.

2.2. Submitting Final Camera-Ready Copy

The final versions of papers accepted for publication should follow the same format and naming convention as initial submissions, except that author information (names and affiliations) should be given. See Section 3.3.2 for formatting instructions.

The footnote, "Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute." must be modified to "*Proceedings of the 42^{nd} International Conference on Machine Learning*, Vancouver, Canada, PMLR 267, 2025. Copyright 2025 by the author(s)."

For those using the **LaTeX** style file, this change (and others) is handled automatically by simply changing `\usepackage{icml2025}` to

`\usepackage[accepted]{icml2025}`

Authors using **Word** must edit the footnote on the first page of the document themselves.

Camera-ready copies should have the title of the paper as running head on each page except the first one. The running title consists of a single line centered above a horizontal rule which is 1 point thick. The running head should be centered, bold and in 9 point type. The rule should be 10 points above the main text. For those using the **LaTeX** style file, the original title is automatically set as running head using the `fancyhdr` package which is included in the ICML 2025 style file package. In case that the original title exceeds the size restrictions, a shorter form can be supplied by using

`\icmltitlerunning{...}`

just before `\begin{document}`. Authors using **Word** must edit the header of the document themselves.

3. Format of the Paper

All submissions must follow the specified format.

3.1. Dimensions

The text of the paper should be formatted in two columns, with an overall width of 6.75 inches, height of 9.0 inches, and 0.25 inches between the columns. The left margin should be 0.75 inches and the top margin 1.0 inch (2.54 cm). The right and bottom margins will depend on whether you print on US letter or A4 paper, but all final versions must be produced for US letter size. Do not write anything on the margins.

The paper body should be set in 10 point type with a vertical spacing of 11 points. Please use Times typeface throughout the text.

3.2. Title

The paper title should be set in 14 point bold type and centered between two horizontal rules that are 1 point thick, with 1.0 inch between the top rule and the top edge of the page. Capitalize the first letter of content words and put the rest of the title in lower case.

3.3. Author Information for Submission

ICML uses double-blind review, so author information must not appear. If you are using LaTeX and the `icml2025.sty` file, use `\icmlauthor{...}` to specify authors and `\icmlaffiliation{...}` to specify affiliations. (Read the TeX code used to produce this document for an example usage.) The author information will not be printed unless `accepted` is passed as an argument to the style file. Submissions that include the author information will not be reviewed.

3.3.1. Self-Citations

If you are citing published papers for which you are an author, refer to yourself in the third person. In particular, do not use phrases that reveal your identity (e.g., "in previous work, we have shown ...").

Do not anonymize citations in the reference section. The only exception are manuscripts that are not yet published (e.g., under submission). If you choose to refer to such unpublished manuscripts, anonymized copies have to be submitted as Supplementary Material via OpenReview. However, keep in mind that an ICML paper should be self contained and should contain sufficient detail for the reviewers to evaluate the work. In particular, reviewers are not required to look at the Supplementary Material when writing their review (they are not required to look at more than the first 8 pages of the submitted document).

3.3.2. Camera-Ready Author Information

If a paper is accepted, a final camera-ready copy must be prepared. For camera-ready papers, author information should start 0.3 inches below the bottom rule surrounding the title. The authors' names should appear in 10 point bold type, in a row, separated by white space, and centered. Author names should not be broken across lines. Unbolded superscripted numbers, starting 1, should be used to refer to affiliations.

Affiliations should be numbered in the order of appearance. A single footnote block of text should be used to list all the affiliations. (Academic affiliations should list Department, University, City, State/Region, Country. Similarly for industrial affiliations.)

Each distinct affiliations should be listed once. If an author has multiple affiliations, multiple superscripts should be placed after the name, separated by thin spaces. If the authors would like to highlight equal contribution by multiple first authors, those authors should have an asterisk placed after their name in superscript, and the term "*Equal contribution" should be placed in the footnote block ahead of the list of affiliations. A list of corresponding authors and their emails (in the format Full Name <email@domain.com>) can follow the list of affiliations. Ideally only one or two names should be listed.

A sample file with author names is included in the ICML2025 style file package. Turn on the `[accepted]` option to the stylefile to see the names rendered. All of the guidelines above are implemented by the LaTeX style file.

3.4. Abstract

The paper abstract should begin in the left column, 0.4 inches below the final address. The heading 'Abstract' should be centered, bold, and in 11 point type. The abstract body should use 10 point type, with a vertical spacing of 11 points, and should be indented 0.25 inches more than normal on left-hand and right-hand margins. Insert 0.4 inches of blank space after the body. Keep your abstract brief and self-contained, limiting it to one paragraph and roughly 4–6 sentences. Gross violations will require correction at the camera-ready phase.

3.5. Partitioning the Text

You should organize your paper into sections and paragraphs to help readers place a structure on the material and understand its contributions.

3.5.1. Sections and Subsections

Section headings should be numbered, flush left, and set in 11 pt bold type with the content words capitalized. Leave 0.25 inches of space before the heading and 0.15 inches after the heading.

Similarly, subsection headings should be numbered, flush left, and set in 10 pt bold type with the content words capitalized. Leave 0.2 inches of space before the heading and 0.13 inches afterward.

Finally, subsubsection headings should be numbered, flush left, and set in 10 pt small caps with the content words capitalized. Leave 0.18 inches of space before the heading and 0.1 inches after the heading.

Please use no more than three levels of headings.

3.5.2. Paragraphs and Footnotes

Within each section or subsection, you should further partition the paper into paragraphs. Do not indent the first line of a given paragraph, but insert a blank line between succeeding ones.

You can use footnotes[1] to provide readers with additional information about a topic without interrupting the flow of the paper. Indicate footnotes with a number in the text where the point is most relevant. Place the footnote in 9 point type at the bottom of the column in which it appears. Precede the first footnote in a column with a horizontal rule of 0.8 inches.[2]

3.6. Figures

You may want to include figures in the paper to illustrate your approach and results. Such artwork should be centered,

[1] Footnotes should be complete sentences.

[2] Multiple footnotes can appear in each column, in the same order as they appear in the text, but spread them across columns and pages if possible.

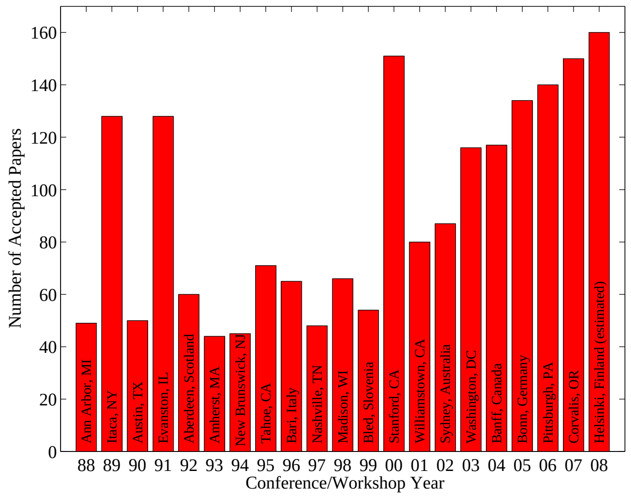

Figure 1. Historical locations and number of accepted papers for International Machine Learning Conferences (ICML 1993 – ICML 2008) and International Workshops on Machine Learning (ML 1988 – ML 1992). At the time this figure was produced, the number of accepted papers for ICML 2008 was unknown and instead estimated.

legible, and separated from the text. Lines should be dark and at least 0.5 points thick for purposes of reproduction, and text should not appear on a gray background.

Label all distinct components of each figure. If the figure takes the form of a graph, then give a name for each axis and include a legend that briefly describes each curve. Do not include a title inside the figure; instead, the caption should serve this function.

Number figures sequentially, placing the figure number and caption *after* the graphics, with at least 0.1 inches of space before the caption and 0.1 inches after it, as in Figure 1. The figure caption should be set in 9 point type and centered unless it runs two or more lines, in which case it should be flush left. You may float figures to the top or bottom of a column, and you may set wide figures across both columns (use the environment `figure*` in LaTeX). Always place two-column figures at the top or bottom of the page.

3.7. Algorithms

If you are using LaTeX, please use the "algorithm" and "algorithmic" environments to format pseudocode. These require the corresponding stylefiles, algorithm.sty and algorithmic.sty, which are supplied with this package. Algorithm 1 shows an example.

Algorithm 1 Bubble Sort

```
Input: data $x_i$, size $m$
repeat
  Initialize $noChange = true$.
  for $i = 1$ to $m - 1$ do
    if $x_i > x_{i+1}$ then
      Swap $x_i$ and $x_{i+1}$
      $noChange = false$
    end if
  end for
until $noChange$ is $true$
```

Table 1. Classification accuracies for naive Bayes and flexible Bayes on various data sets.

DATA SET	NAIVE	FLEXIBLE	BETTER?
BREAST	95.9± 0.2	96.7± 0.2	$\surd$
CLEVELAND	83.3± 0.6	80.0± 0.6	$\times$
GLASS2	61.9± 1.4	83.8± 0.7	$\surd$
CREDIT	74.8± 0.5	78.3± 0.6	
HORSE	73.3± 0.9	69.7± 1.0	$\times$
META	67.1± 0.6	76.5± 0.5	$\surd$
PIMA	75.1± 0.6	73.9± 0.5	
VEHICLE	44.9± 0.6	61.5± 0.4	$\surd$

3.8. Tables

You may also want to include tables that summarize material. Like figures, these should be centered, legible, and numbered consecutively. However, place the title *above* the table with at least 0.1 inches of space before the title and the same after it, as in Table 1. The table title should be set in 9 point type and centered unless it runs two or more lines, in which case it should be flush left.

Tables contain textual material, whereas figures contain graphical material. Specify the contents of each row and column in the table's topmost row. Again, you may float tables to a column's top or bottom, and set wide tables across both columns. Place two-column tables at the top or bottom of the page.

3.9. Theorems and such

The preferred way is to number definitions, propositions, lemmas, etc. consecutively, within sections, as shown below.

Definition 3.1. A function $f : X \to Y$ is injective if for any $x, y \in X$ different, $f(x) \neq f(y)$.

Using Theorem 3.1 we immediate get the following result:

Proposition 3.2. *If f is injective mapping a set X to another set Y, the cardinality of Y is at least as large as that of X*

Proof. Left as an exercise to the reader. □

Theorem 3.3 stated next will prove to be useful.

Lemma 3.3. *For any $f : X \to Y$ and $g : Y \to Z$ injective functions, $f \circ g$ is injective.*

Theorem 3.4. *If $f : X \to Y$ is bijective, the cardinality of X and Y are the same.*

An easy corollary of Theorem 3.4 is the following:

Corollary 3.5. *If $f : X \to Y$ is bijective, the cardinality of X is at least as large as that of Y.*

Assumption 3.6. The set X is finite.

Remark 3.7. According to some, it is only the finite case (cf. Theorem 3.6) that is interesting.

3.10. Citations and References

Please use APA reference format regardless of your formatter or word processor. If you rely on the LaTeX bibliographic facility, use `natbib.sty` and `icml2025.bst` included in the style-file package to obtain this format.

Citations within the text should include the authors' last names and year. If the authors' names are included in the sentence, place only the year in parentheses, for example when referencing Arthur Samuel's pioneering work. Otherwise place the entire reference in parentheses with the authors and year separated by a comma. List multiple references separated by semicolons. Use the 'et al.' construct only for citations with three or more authors or after listing all authors to a publication in an earlier reference.

Authors should cite their own work in the third person in the initial version of their paper submitted for blind review. Please refer to Section 3.3 for detailed instructions on how to cite your own papers.

Use an unnumbered first-level section heading for the references, and use a hanging indent style, with the first line of the reference flush against the left margin and subsequent lines indented by 10 points. The references at the end of this document give examples for journal articles, conference publications, book chapters, books, edited volumes, technical reports, and dissertations.

Alphabetize references by the surnames of the first authors, with single author entries preceding multiple author entries. Order references for the same authors by year of publication, with the earliest first. Make sure that each reference includes all relevant information (e.g., page numbers).

Please put some effort into making references complete, presentable, and consistent, e.g. use the actual current name of authors. If using bibtex, please protect capital letters of names and abbreviations in titles, for example, use {B}ayesian or {L}ipschitz in your .bib file.

Impact Statement

Authors are **required** to include a statement of the potential broader impact of their work, including its ethical aspects and future societal consequences. This statement should be in an unnumbered section at the end of the paper (co-located with Acknowledgements – the two may appear in either order, but both must be before References), and does not count toward the paper page limit. In many cases, where the ethical impacts and expected societal implications are those that are well established when advancing the field of Machine Learning, substantial discussion is not required, and a simple statement such as the following will suffice:

"This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here."

The above statement can be used verbatim in such cases, but we encourage authors to think about whether there is content which does warrant further discussion, as this statement will be apparent if the paper is later flagged for ethics review.

4. Problem Definition

The goal of our project was to create a model which forecasts the future points of a time-series in a zero-shot manner. Concretely we want to build a model, that takes in k previous windows of length L of a time-series and then predicts the next $k + 1$ window of length L. The model should also work in a zero-shot manner, meaning it should work without requiring any fine-tuning on the specific data it will be used on. To accomplish this we constructed a synthetic training dataset, build to cover a wide range of different time-series for the model to learn in order to facilitate the zero-shot use of the model. We also added artificial noise into our time-series data to make the model more robust and usable for real world data. Our model consists of two separate networks, one for encoding the k windows and a second one for predicting the window $k + 1$ from the encodings.

5. Methods

In this section we will look at the two main architectures that our model is based on. We will first look at DeepONet (Lu et al., 2021), which encodes out time-series windows and fits a function on the noisy time points, in Section 5.1 and afterwards at FIM (Seifner et al., 2024), which predicts the future encoding, in Section 8.1.

5.1. DeepONet

Deep operator network (DeepONet)(Lu et al., 2021) is a neural network architecture designed to learn nonlinear operators more accurately and efficiently than standard fully-connected networks. DeepONet consists of two subnetworks, a branch net and a trunk net. The branch net takes in m fixed values of the input function $f(t)$ and encodes them while the trunk net takes in the time point t^* for the output function we want to predict and encodes it. Both networks return a p-dimensional encoding and to obtain the output function value at our time point t^*, the scalar product between both encodings is calculated.

5.2. FIM

FIM(Seifner et al., 2024) is a model originally designed for the interpolation of time series displaying temporal missing patterns. Meaning we have our k embeddings from $1, 2, ..., q-1, q+1, ..., k$ with the embedding q missing. The task of FIM was then to predict the embedding for window q from the other embeddings. In our work we used the core ideas of FIM but for forecasting the future embedding $k+1$. For this FIM takes in the embeddings $h_1, .., h_k$ along with local scale embeddings $s_1, .., s_k$. These scale embeddings are created by saving the normalization values, used to normalize the time-series data for each respective window, and feeding them through a MLP to obtain an embedding. Both embeddings then get concatenated and fed through a transformer and summary network to obtain the embedding h_{k+1}. With this embedding we can use the trunk net from DeepONet to obtain a function for future time points t^*.

6. Synthetic Dataset

For training and testing our models we created two synthetic datasets (one for each model) containing different time-series. For the DeepONet model we created a dataset containing 100K random one-dimensional, target interpolation functions $f(t)$ on the interval $[0, 1]$, defined on a fine grid of 128 points. We sampled them from Gaussian Processes with an RBF kernel and mean 0. The kernel has two hyperparameters, the scale which controls the scale at which variations take place and the variance. The scale is drawn randomly from a beta distribution and scaled by 0.1 to increase the variety of the vector fields in our dataset. Depending on weather or not we are creating a train or test dataset the alpha and beta values for the beta distributions are randomly chosen from either $[1.0, 2.0, 5.0]$ or $[0.5, 3.25, 7.8]$ respectively. The variance for the kernel is drawn uniformly from the interval $[0.5, 2]$. we then sample random observations grid points t on our functions, which become the input for our model. We sample a random subset between "min=50" and "max=90" points from the 128 grid points and pad the rest with 0's. We also create a binary mask for each observation grid, which track which indices are observed versus unobserved, that gets fed into our model along with the points. Half of our observation points are sampled regular and the other half irregular (in time). Since data is noisy in real world applications we add gaussian noise to the observations with a standard deviation drawn from $\mathcal{N}(0, 0.1)$ and mean 0 after first normalizing our functions to $[0, 1]$.

7. Model architecture

Next, we proceed with a detailed description of the architectures for both the FIM-l and FIM models. The FIM-l model serves as an operator, aiming to learn the underlying function from noisy samples, while the FIM model is designed as a forecasting framework. Our description closely follows the original implementations of both architectures, as outlined in (Seifner et al., 2024).

7.1. FIM-ℓ

The primary aim of the FIM-ℓ model is to learn the underlying function $f(\tau)$ that has been augmented by noise to generate the observed time series (y_i, τ_i). The model should allow querying interpolated values of the underlying function $f(\tau)$ at arbitrary time points, including those not present in the observed data. We can therefore think of FIM-ℓ as a learned *neural interpolation operator* that maps the observed data into a continuous function space. To achieve this, we will leverage the ideas and architecture proposed by DeepONet (Lu et al., 2021). Given our noisy input sequence $(y_1, \tau_1), \dots, (y_l, \tau_l)$, with observation values $y_i \in \mathbb{R}$ and ordered observation times $\tau_i \in \mathbb{R}^+$, as well as query points t_i, we define two feedforward neural network (FFN) embedding networks, ϕ_0^θ and ϕ_1^θ, to transform both the observed values and time points into an embedded representation:

$$\hat{y_i} = \phi_0^\theta(y_i), \quad \hat{t_i} = \phi_1^\theta(t_i).$$

We then proceed by concatenating both components to obtain the individual observation embeddings:

$$\mathbf{y_i^\theta} = \text{Concat}(\hat{y_i}, \hat{t_i}).$$

Following the work of DeepONet, we define a *branch net*-equivalent network consisting of a transformer-encoder network (**?**), denoted as ψ_0^θ, and a multilayer perceptron (MLP), denoted as ϕ_3^θ. Together, these form

$$\mathbf{u}^\theta = \phi_3^\theta\big(\psi_0^\theta(\mathbf{y_1^\theta}, \dots, \mathbf{y_l^\theta})\big).$$

Finally, to generate a sequence-length-agnostic embedding, we take $\mathbf{u}^\theta$ from the branch network and feed it into a Multi-Head Attention () summary block λ_0^θ, where $\mathbf{u}^\theta$ serves as the *keys* and *values*, and a *learnable* vector q_{θ^*} is used as the *query*. The attention calculation is defined as

$$\mathbf{h}^\theta = \text{softmax}\left(\frac{q_{\theta^*} K^\top}{\sqrt{d_k}}\right) V = \lambda_0^\theta(\mathbf{u}^\theta),$$

where $K = \mathbf{u}^\theta$ are the keys, $V = \mathbf{u}^\theta$ are the values, and d_k is the dimensionality of the keys.
Next, we define our *trunk net*-equivalent network. We begin by introducing a separate embedding network, ϕ_4^θ, for the query points t. Additionally, we define another MLP, ϕ_5^θ. The final trunk net output, $\mathbf{t}^\theta$, is then obtained as

$$\mathbf{t}^\theta = (\phi_5^\theta \circ \phi_4^\theta)(t).$$

To finally obtain the interpolated values of the learned underlying function at the query points t, we define a final MLP, ϕ_7^θ, such that

$$\mathbf{y}(t) = \phi_7^\theta\big(\text{Concat}(\mathbf{h}^\theta, \mathbf{t}^\theta)\big),$$

where $\mathbf{y}(t)$ represents the learned underlying function given our noisy observation values.

7.2. FIM

We now proceed by utilizing the learned representations $\mathbf{h}^\theta$ of each local function window to predict the values of the time series at arbitrary points within the next, previously unseen window. Starting from the beginning, we receive a noisy time sequence $(y_1, \tau_1), \dots, (y_l, \tau_l)$. We then split these values into $K = 5$ windows, such that for window S_j, we have

$$S_{ji} = (y_{\alpha+i}, \tau_{\alpha+i}), \quad \alpha = \sum_{l=1}^{j-1} w_l,$$

where w_l is the number of observations in window $l \leq K-1$. Additionally, for each of these windows, we construct their local scale characteristics $s_l \in \mathbb{R}^9$ (??), which are fed into an embedding layer σ_0^ω, defined as

$$\hat{s}_l = \sigma_0^\omega(s_l).$$

We then pass each window of observations into the pretrained embedding layers of the FIM-ℓ model. Specifically, we define:

$$\mathbf{y}_i^j = \text{Concat}\big(\phi_0^\theta((S_{ji})_1), \phi_1^\theta((S_{ji})_2)\big), \quad \text{j} \leq K-1, i \leq w_j.$$

We proceed by passing these $\mathbf{y}^j$ into the branch network of the FIM-ℓ, resulting in

$$\mathbf{h}_j = (\lambda_0^\theta \circ \phi_3^\theta \circ \psi_0^\theta)\big(\mathbf{y}_1^j, \dots, \mathbf{y}_{w_j}^j\big).$$

After extracting the local embeddings for each of the $K-1$ windows, we proceed to reconstruct the K-th window. To achieve this, we first concatenate each local-scale embedding $\hat{s}_j$ with the observation embeddings and feed them into a Transformer encoder block ψ_1^ω. This is again followed by an attention-based summary network λ_1^ω, which generates the final embedding for the K-th window

$$\mathbf{h}_K^* = (\eta_0^\omega \circ \lambda_1^\omega \circ \psi_1^\omega)\Big((\mathbf{h}_1, \hat{s}_1), \dots, \eta_0^\omega(\mathbf{h}_{K-1}, \hat{s}_{K-1})\Big).$$

Due to the concatenation of the observation and scale embeddings, the feature dimension is now doubled compared to the original embedding size. However, the frozen projection network of FIM-ℓ expects inputs in the original embedding dimension. To address this, we utilize an extractor network η_0^σ, which transforms the output of the summary network λ_1^ω back to the dimension expected by the FIM-ℓ projection layer.

To generate the final predictions for the K-th window, we utilize the embedding and trunk networks of the pretrained FIM-ℓ to predict the function values at the query points t. This is expressed as

$$\mathbf{y}(t) = \phi_7^\theta\big(\text{Concat}(\mathbf{h}_K^*, (\phi_5^\theta \circ \phi_4^\theta)(t))\big).$$

8. Model Training

In this section, we provide the necessary information to ensure the reproducibility of our work. Specifically, we outline the detailed structure of the aforementioned MLPs and discuss all relevant hyperparameters for both the model and the optimization algorithms.

8.1. FIM-ℓ Training

The implementation of our previously defined FIM-ℓ architecture is described in 8.1. Here, we use d_{model} to denote the embedding dimension. In our setting, we set $d_{\text{model}} = 256$ and $n_{\text{heads}} = 4$. Since *LeakyReLU* is shift-invariant, the bias term in the linear layer can be omitted if it is followed by a *LayerNorm*, as the *LayerNorm* neutralizes any bias introduced by the preceding layer. We train the model with a batch size of 128, using the *AdamW* optimizer with the following hyperparameters: β-values $(\beta_1, \beta_2) = (0.9, 0.999)$, $\epsilon = 10^{-8}$, and a weight decay of 0.01. The training is performed for 20 epochs, which took approximately one hour. Additionally, we employ an *Inverse Square Root Learning Rate* (*InverseSquareRootLR*)(**?**) scheduling strategy, with 100 warm-up steps, an initial learning rate of 10^{-4}, and a minimum learning rate of 10^{-5}.
To save memory and computational resources, we utilize the PyTorch *Automatic Mixed Precision* package, which trains the model in mixed precision. Specifically, it selects half-precision data types (*bfloat16* in our case) for operations it deems suitable. This approach enables the model to leverage the highly optimized NVIDIA Tensor Cores, maximizing performance during matrix operations.
For our loss computations, we use the standard *Mean Squared Error* (MSE) between the predicted outputs of the model and the precomputed ground truth. Before performing the optimizer update step, we apply gradient clipping to ensure that the gradient norm does not exceed the length of a unit vector. This stabilizes training by limiting the size of each gradient step during optimization.

We then provide the model with the noisy observation sequence, observation time points, query points, and the branch mask. The branch mask is then utilized by both the Transformer encoder ψ_0^θ and the summary network λ_0^θ as the padding mask.

Component	**Details**
Branch Embedding ϕ_0^θ	Linear(1, d_{model})
Branch Embedding ϕ_1^θ	Linear(1, d_{model})
Trunk Embedding ϕ_4^θ	Linear(1, d_{model})
Branch Encoder Input	Concatenate embeddings of y and t
Branch Encoder ψ_0^θ	Transformer Encoder (6 layers, $2d_{\text{model}}$, n_{heads})
Branch MLP ϕ_3^θ	Linear($2d_{\text{model}} \to d_{\text{model}}$), LeakyReLU, LayerNorm
Learnable Query	Parameter tensor of shape (1, d_{model})
Branch Attention λ_0^θ	Multihead Attention (d_{model}, heads=1)
Trunk MLP ϕ_5^θ	4x Linear($d_{\text{model}} \to d_{\text{model}}$), LeakyReLU, LayerNorm
Combine Outputs	Concatenate outputs of Branch Attention and Trunk MLP
Final Projection ϕ_7^θ	5x Linear ($2d_{\text{model}} \to 1$), LeakyReLU, LayerNorm

Table 2. FIM-ℓ architecture implementation

Component	**Details**
Pretrained FIM-ℓ	$n_{\text{heads-fim-}\ell}$, d_{model} (Frozen)
Local Scale Embedding σ_0^ω	Linear(9, d_{model})
Combine Outputs	Concatenate outputs of Pretrained FIM-ℓ and Local Scale Embedding
Transformer Encoder ψ_1^ω	8 layers, $2d_{\text{model}}$, n_{heads}
Learnable Query	Parameter tensor of shape (1, $2d_{\text{model}}$)
Summary Attention λ_1^ω	Multihead Attention ($2d_{\text{model}}$, n_{heads})
Extractor Network η_0^ω	Sequential: Linear($2d_{\text{model}}$, $4d_{\text{model}}$) LeakyReLU, LayerNorm Linear($4d_{\text{model}}$, $2d_{\text{model}}$) LeakyReLU, Linear($2d_{\text{model}}$, d_{model})
Trunk Embedding ϕ_4^θ	Reused from FIM-ℓ (Frozen)
Trunk MLP ϕ_5^θ	Reused from FIM-ℓ (Frozen)
Combine Outputs	Concatenate outputs of Trunk Network and Summary Network
Final Projection ϕ_7^θ	Reused from FIM-ℓ (Frozen): 5x Linear ($2d_{\text{model}} \to 1$), LeakyReLU, LayerNorm

Table 3. FIM Architecture Overview

8.2. FIM Training

The implementation of the FIM network that we defined is detailed in Table 8.2. We set $d_{\text{model}} = 256$ and $n_{\text{heads}} = 8$. Regarding the optimizer and learning rate strategy, we use the same settings as described previously, along with automatic mixed precision training. The loss function remains the *Mean Squared Error* (MSE), and the gradients are clipped to ensure their norm does not exceed the length of a unit vector. We again provide the model with the noisy observation sequence, observation time points, query points, and the branch mask. However, instead of treating a single window as one data point, each example now comprises all local windows of the global function. We then continue training for 70 epochs, which takes approximately 4 hours. All model training was conducted on an NVIDIA RTX 3070 GPU, equipped with 8GB of GDDR6 VRAM.

9. Experiments

In this section, we discuss additional experiments conducted on the model's architecture and the construction of the loss function to achieve higher prediction accuracy. The impact of each of these approaches will be presented later in Section **??**.

9.1. Learned Positional Encoding for FIM-ℓ Embeddings

In this approach, we hypothesize that the model may benefit from additional positional encoding for the attention-based summary network λ_1^ω. We define

$$\mathbf{p} = \psi_1^\omega\Big((\mathbf{h}_1, \hat{s}_1), \ldots, \eta_0^\omega(\mathbf{h}_{K-1}, \hat{s}_{K-1})\Big), \quad \mathbf{p} \in \mathbb{R}^{(K-1)\times d_{\text{model}}}.$$

as the output from the FIM transformer encoder network. In our standard implementation, $\mathbf{p}$ is passed to the summary attention network λ_1^ω, which, by default, lacks a sense of order among these embeddings, aside from the local scale statistics of each window. We hypothesize that providing positional information may help this layer better identify the order of embeddings, enabling more informed predictions of the K-th embedding.
To address this, we introduce an additional parameter vector $\mathbf{z} \in \mathbb{R}^{(K-1)\times d_{\text{model}}}$, which serves as a learnable positional encoding. We add $\mathbf{z}$ elementwise to $\mathbf{p}$ before passing the result to λ_1^ω, as follows

$$\mathbf{h}_K^* = (\eta_0^\omega \circ \lambda_1^\omega)\Big(\mathbf{p} + \mathbf{z}\Big).$$

9.2. Similarity loss between FIM-ℓ ground truth and predicted embeddings.

Another strategy is the usage of the ground truth $\mathbf{h}_K$ for enabling more accurate predictions. The idea is to align the predicted $\mathbf{h}_K^*$ and the actual ground truth $\mathbf{h}_K$ such that they are as close as possible to each other. We can do this by adding an additional term to the model's loss function, which incentivizes it to form predictions $\mathbf{h}_K^*$ that are close to the FIM-ℓ predicted $\mathbf{h}_K$. A suitable metric for this is the *cosine similarity*, defined as

$$\text{CosSim}(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a}^T\mathbf{b}}{\|\mathbf{a}\|\|\mathbf{b}\|}, \quad \mathbf{a}, \mathbf{b} \in \mathbb{R}^n.$$

$$\forall \mathbf{a}, \mathbf{b} \in \mathbb{R}^n \setminus \{\mathbf{0}\}, \quad \text{CosSim}(\mathbf{a}, \mathbf{b}) \in [-1, 1].$$

Geometrically, the *cosine similarity* represents the cosine of the angle between two vectors. Specifically, $\text{CosSim}(\mathbf{a}, \mathbf{b}) = 0$ indicates orthogonality, while $\text{CosSim}(\mathbf{a}, \mathbf{b}) = 1$ and $\text{CosSim}(\mathbf{a}, \mathbf{b}) = -1$ correspond to vectors pointing in the same and opposite directions, respectively.
To give the network a better chance of transforming $\mathbf{h}_K^*$, we introduce an additional 4-layer MLP called λ_2^ω. We then calculate the cosine similarity (CosSim) between the output of λ_2^ω and $\mathbf{h}_K$.
We now define our new loss function $\mathcal{L}(y, \hat{y}, \mathbf{h}_K^*, \mathbf{h}_K)$, which incorporates both prediction accuracy and alignment between the predicted and ground-truth representations. The loss function is given by

$$\mathcal{L}(y, \hat{y}, \mathbf{h}_K^*, \mathbf{h}_K) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \beta \cdot |\text{CosSim}(\lambda_2^\omega(\mathbf{h}_K^*), \mathbf{h}_K) - 1|.$$

where y denotes the ground truth values of the target variable, and $\hat{y}$ represents the corresponding predicted values by the model. We choose $\beta = 0.2$ to ensure that the optimization does not focus too aggressively on aligning $\mathbf{h}_K^*$ and $\mathbf{h}_K$.

10. Results

We will now discuss our findings regarding both the baseline architecture and the performance of our additional experiments. For evaluation, we utilize both our validation set and the *ETTh1* dataset, which comprises real-world time series data from multiple domains.

10.1. Findings on Validation Set

On our validation set, we evaluate the performance of our models with the *Mean Absolute Error* (MAE) defined as

$$\text{MAE}(y, \hat{y}) = \frac{1}{n}\sum_{1}^{n}|y_i - \hat{y}_i|.$$

In Table 4, we present the results on our validation set for multiple instances of our model. We abbreviate our Standard model with the capital letter S, as well as PE for positional encoding and CSL for cosine-similarity loss. Unfortunately, as shown in the table, neither the addition of positional encoding to the FIM-ℓ embeddings in the summary network nor the introduction of the cosine loss yielded any significant benefit. However, we observed a slightly lower training loss for the PE variant.

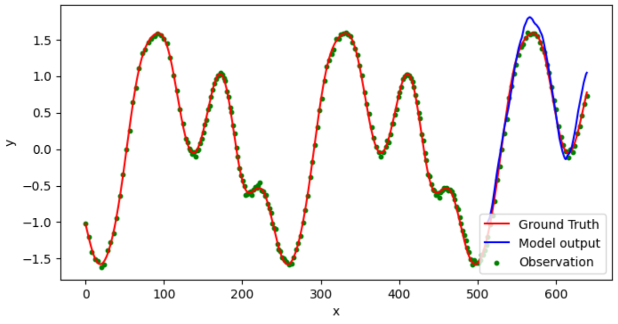

Figure 2. Your caption here.

Looking at the plots of the predictions generated by our model on the validation set, we observed that the model performs well on data with high periodicity (see Figure 2). Since the functions and the parameters for the Gaussian process are sampled randomly, it is possible to encounter functions with less clear patterns (see Figure 3). These functions pose a challenge for the model, as it cannot anticipate their behavior in advance. This results in the model learning the average course of such functions. We postulate that this behavior negatively impacts the model's performance on functions where distinct patterns are present. It's therefore crucial to have checks in place to ensure that the model is only trained on data that exhibits clear patterns or aligns with the desired characteristics of the target task. This may help prevent the model from overfitting to noise or learning irrelevant trends.

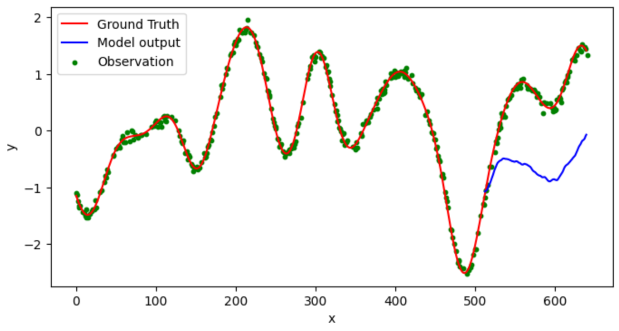

Figure 3. Your caption here.

10.2. Findings on ETTh1

ETTh1 is a time series dataset containing data from various domains, such as oil temperature and electrical charge. We evaluate our model with a time horizon of 96, which corresponds to capturing 480 data points. These are split into $K = 5$ windows, with the task being to predict the 5th window. The five windows are then shifted by one to the right and evaluated again. The same procedure is applied for the time horizon of 192. We evaluate our different

	S	PE	PE + CSL	TimesFM
Validation	0.576	0.579	0.585	-
ETTh1 - 96	0.966	0.968	0.928	0.43*
ETTh1 - 192	1.0904	1.089	1.043	0.43*

Table 4. MAE results for three model variants and TimesFM on two datasets.
* TimesFM () only reports the average score over both time horizons. Additionally, they use only 4 datasets from ETTh1, which they do not disclose.

model variants on the entire ETTh1 dataset in a zero-shot manner, meaning that we evaluate on all sub-datasets contained within ETTh1 and calculate the average performance. Additionally, we compare our model to the one proposed by TimesFM (), using their accuracy score as a benchmark. As shown in Table 4, positional encoding does not appear to offer a significant benefit over the standard model. However, the PE + CSL approach provides slightly better performance. We also observe that the zero-shot performance of TimesFM on ETTh1-96 is much better than that of our current model. It is important to note, however, that TimesFM has been trained on a variety of data, both real-world and synthetic, and is approximately an order of magnitude larger in terms of parameter count. We observe the same ordering of models for a horizon window of 192, where the PE + CSL version performs slightly better than the vanilla and PE versions.

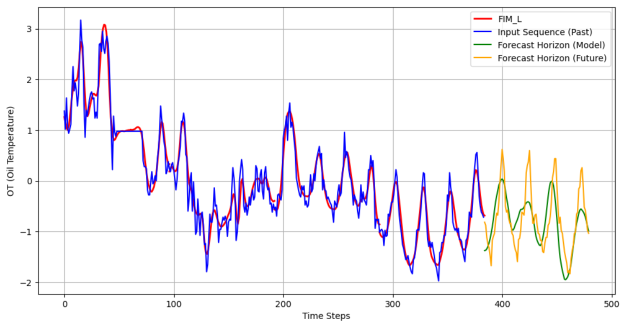

Figure 4. Example of good performance on the oil temperature dataset with horizon $= 96$, using the PE + CSL model version.

Finally, we present two examples from the ETTh1 dataset using our model to showcase both well-performing and poorly-performing cases. In Figure 4, the model successfully predicts the horizon, as the data follows a very predictable pattern. However, in Figure 5, we observe that the model struggles to accurately predict the horizon probably due to the lack of a very clear pattern in the data.

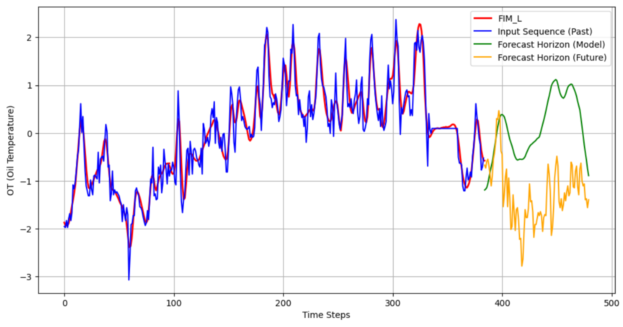

Figure 5. Example of bad performance on the oil temperature dataset with horizon $= 96$, using the PE + CSL model version.

This suggests that our networks would likely benefit significantly from training on more diverse and real-world datasets, allowing them to learn patterns that are more subtle than those generated by a Gaussian process with a periodic kernel.

11. Conclusion

References

Lu, L., Jin, P., Pang, G., Zhang, Z., and Karniadakis, G. E. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, March 2021. ISSN 2522-5839. doi: 10.1038/s42256-021-00302-5. URL `http://dx.doi.org/10.1038/s42256-021-00302-5`.

Seifner, P., Cvejoski, K., Körner, A., and Sánchez, R. J. Zero-shot imputation with foundation inference models for dynamical systems. In *Submitted to The Thirteenth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=NPSZ7V1CCY`. under review.

A. You *can* have an appendix here.

You can have as much text here as you want. The main body must be at most 8 pages long. For the final version, one more page can be added. If you want, you can use an appendix like this one.

The `\onecolumn` command above can be kept in place if you prefer a one-column appendix, or can be removed if you prefer a two-column appendix. Apart from this possible change, the style (font size, spacing, margins, page numbering, etc.) should be kept the same as the main body.