November 2025
Volume 1
Issue 3

DOI: 10.48308/ijce.2025.241830.1016

Interdisciplinary Journal of Civil Engineering
Shahid Beheshti University

Enhanced Multimodal Deep Learning Framework for Accurate El Niño and La Niña Forecasting

Authors:

Saghar Ganji[1], Alireza Hassani[2], Ahmad Reza Labibzadeh[3], Mohammad Naisipour[4, *]

Abstract

Reliable long-lead forecasting of the El Niño–Southern Oscillation (ENSO) remains a long-standing challenge in climate science. The previously developed Multimodal ENSO Forecast (MEF) model uses 80 ensemble predictions by two independent deep learning modules: a 3D Convolutional Neural Network (3DCNN) and a time-series Informer module. In their approach, outputs of the two modules are combined using a weighting strategy wherein one is prioritized over the other as a function of global performance. Separate weighting or testing of individual ensemble members did not occur, however, which may have limited the model from optimizing the use of high-performing but spread-out forecasts. In this study, we propose a novel framework that employs graph-based analysis to implicitly find the best ensemble. By constructing an undirected graph whose vertices are ensemble outputs and whose weights on edges measure similarity, we identify and cluster the best prediction. This method improves the forecast skill through noise removal and emphasis on ensemble coherence. Interestingly, our graph-based selection shows robust statistical characteristics among top performers, offering new ensemble behavior insights. The approach is model-agnostic too, suggesting that it can be applied directly to other forecasting models with gargantuan ensemble outputs, such as statistical and physical models.

Keywords: El Niño, La Niña, Forecast, MEF, Ensemble Selection, GNN

1. Department of Computer Engineering, Shiraz University of Technology, Shiraz, Iran

2. Jacksonville University, Jacksonville, Florida, USA

3. Department of Computer Engineering, Sharif University of Technology, Tehran, Iran

4. Department of Civil Engineering, Faculty of Engineering, University of Zanjan, Zanjan, Iran

*Corresponding Author: m.naisipour@znu.ac.ir

1. Introduction

Long-lead-time prediction of the El Niño–Southern Oscillation (ENSO) is one of the central problems of climate science since the system is inherently nonlinear, is dynamically chaotic, and involves complex teleconnections (McPhaden, Zebiak, & Glantz, 2006; L'Heureux et al., 2020). Successful prediction of ENSO is of crucial practical value due to its strong global impacts on the weather, agriculture, water resources, and socio-economic activity (Cai, Borlace, & Lengaigne, 2014; Cai, McPhaden, & Grimm, 2020). Even though tremendous advances have been achieved in model innovations, it is still not reliable to predict the behavior of ENSO more than one year in advance, especially after 2000, when climate variability has been increasingly unpredictable (Cai, Borlace, & Lengaigne, 2014; Wang, Cheng, Zhang, & Yu, 2023; Scaife et al., 2010).

The advent of artificial intelligence has precipitated a transformative epoch, reshaping economic ecosystems while simultaneously redefining computational paradigms and capabilities (Hassani & Javadi, 2025a; Hassani & Javadi, 2025b). The recent breakthroughs in deep learning have introduced new techniques for nonlinear and spatio-temporal pattern extraction in ocean-atmosphere data (Ham, Kim, & Luo, 2019; Luo, Masson, Behera, & Yamagata, 2008). This is achieved in the Multimodal ENSO Forecasting (MEF) model, wherein two independent modules are 3D Convolutional Neural Networks (3DCNN) and a time-series forecasting module, namely Informers, both having 40 ensemble outputs. The original baseline MEF approach employed a weighted fusion framework, which combined the module outputs based on their aggregate uncertainties (Naisipour, Saeedpanah, & Adib, 2024). However, this method did not blend the individual strengths of ensemble members nor explore the inherent structure within the ensemble space. Consequently, extremely precise predictions were undermined in the process of averaging, while less precise outputs could have been assigned undue influence (Naisipour, Saeedpanah, & Adib, 2025a).

To avoid this shortcoming, we present a GNN-augmented MEF methodology whereby not only the ensemble members' similarity is also expressed explicitly as a graph structure. The ensemble runs are nodes, and edge weights are calculated based on measures of similarity such as RMSE and correlation. With Graph Neural Networks (GNNs), we detect clusters of stable and accurate predictions from the ensemble. This enables us to choose a subset of the most structurally robust and statistically improved outputs, which are then combined to make the final prediction. This graph model has a few major innovations. It enables noise filtering by removing unstable or inconsistent members of the ensemble. It improves the interpretability of model predictions by considering model performance in the framework of structural similarity. It enhances forecast ability, particularly with long lead times, by using internally consistent subsets. It is model-independent and transferable to other ensemble-based prediction systems other than ENSO.

In addition, our graph-theoretic method extracts emergent trends and statistical features from the most successful forecasts, demonstrating that GNNs are not just an ensemble filter but also have the potential to unveil intrinsic ensemble dynamics. These results lay the foundation for the next generation of ensemble post-processing, where not only performance metrics (Naisipour, Saeedpanah, & Adib, 2025b) but also structural consistency across the ensemble inform selection. Our method also has the promise to benefit other climate models and prediction systems generating large ensembles. In contrast to naive average or ad hoc weighting, the proposed method can help determine the best possible forecast out of a collection of outputs, leading to more solid, consistent, and interpretable predictions.

This paper presents the design, implementation, and evaluation of the GNN-augmented MEF model. We demonstrate that graph-theoretic ensemble filtering improves prediction performance, especially for post-2000 climate regimes. The remainder of this paper is organized as follows: Section 2 presents the proposed GNN-based MEF model architecture; Section 3 presents experimental results and discussion; and Section 4 concludes with a discussion on key findings and future research.

2. Methodology

The original MEF model comprises three modules: the 3DCNN, Informers, and the Adaptive Ensemble Module (AEM). We incorporated a Graph Neural Network (GNN) module to extract similarities among multiple model runs that replaces the AEM module. This model integrates the 3DCNN architecture with Transformer methodologies via a fusion network to enable predictions of ENSO events extending beyond one year. The subsequent sections provide a detailed description of each module.

2.1. Input and Label Data

The input for the 3DCNN module consists of a 3D tensor with dimensions of 72 × 24 × 12, wherein each horizontal pixel corresponds to a spatial resolution of 5° × 5° on the Earth's surface, and the dimension of 12 represents the number of temporal frames. This module utilizes a sequence of frames comprising Sea Surface Temperature (SST) and vertically averaged oceanic temperature anomaly maps (Heat Content) within the upper 300 meters as primary predictors. These anomaly maps encompass a geographical region extending from 0° to 360° longitude and from 55° South to 60° North latitude. The target variable, or predictand, is the Niño 3.4 Index, which is defined as the average surface SST anomaly calculated over the longitudinal range of 120° to 170° West and the latitudinal range of 5° South to 5° North. The Niño 3.4 Index is widely recognized as the predominant metric for characterizing El Niño and La Niña events. The MEF model is trained using CMIP5 data and validated with GODAS observational data, ensuring consistency between simulation outputs and real-world dynamics.

Given that the model is data-driven, the availability of sufficient and reliable data is paramount. Consequently, we procured approximately one terabyte of raw data from 21 CMIP5 models, along with GODAS observational data, sourced from reputable repositories, as detailed in Table 1. Due to the varying spatial resolutions of these datasets, we processed them to generate SST and Heat Content (HC) anomaly maps at a resolution of 24x72, alongside the Niño 3.4 Index, which served as the labels. The integrity of the resultant datasets was validated against data provided by Ham et al (Ham, Kim, & Luo, 2019). to ascertain the reliability of the input data.

The 3DCNN module was trained using CMIP5 data spanning the years 1875 to 1975, with validation conducted on GODAS data from 2000 to 2017. The 25-year interval between the training and validation datasets mitigates the influence of oceanic memory effects.

Table 1. List of data and their repositories

Title	Application	Repository Address
CMIP5	Training the 3DCNN model	https://esgf-node.llnl.gov/projects/cmip5/
GODAS (Niño 3.4 Index)	Validation of model predictions	https://www.esrl.noaa.gov/psd/data/gridded/data.godas.html

In addition, Table 2 presents the details of the 21 CMIP5 models used for training the proposed MEF model, including modeling centers and time coverage.

2.2. Three-Dimensional Convolutional Neural Networks

With the emergence of the big data era, machine learning has played a pivotal role in advancing human knowledge across diverse fields by uncovering complex structures and patterns that are often challenging for humans to discern. In particular, Convolutional Neural Networks (CNNs) have demonstrated exceptional efficacy in analyzing multidimensional arrays with spatial structures, such as in object recognition within color images. Consequently, CNNs are well-suited for detecting relationships between predictor fields and precursor patterns. CNNs are specifically designed to automatically learn spatial hierarchies of features through multiple components, including convolutional filters, pooling operations, and fully connected layers. The formula and architecture of the 3DCNN module utilized in the MEF model for classifying input videos are presented in Equation (1) and illustrated in Figure 1, respectively.

Table 2. List of 21 CMIP5 models used in this study, with modeling centers and time spans

No.	ID	Institute	Period
1	BCC-CSM1.1-m	Beijing Climate Center, China Meteorological Administration	1850-1975
2	CanESM2	Canadian Centre for Climate Modelling and Analysis	1850-1975
3	CCSM4	National Center for Atmospheric Research	1850-1975
4	CESM1-CAM5	Community Earth System Model Contributors	1850-1975
5	CMCC-CM	Centro Euro-Mediterraneo per I Cambiamenti Climatici	1850-1975
6	CMCC-CMS		
7	CNRM-CM5	Centre National de Recherches Meteorologiques /Centre Europeen Recherche et Formation Avancee en Calcul Scientifique	1850-1975
8	CSIRO-Mk3-6-0	Commonwealth Scientific and Industrial Research Organization in Collaboration with Queensland Climate Change System of Excellence	1850-1975
9	FIO-ESM	The First Institute of Oceanography, SOA, China	1850-1975
10	GFDL-ESM2G	NOAA Geophysical Fluid Dynamics Laboratory	1861-1975
11	GISS-E2-H	NASA Goddard Institute for Space Studies	1850-1975
12	HadGEM2-AO	National Institute of Meteorological Research/Korea Meteorological Administration	1860-1975
13	HadCM3	Met Office Hadley Centre (additional HadGEM2-ES realizations contributed by Instituto Nacional de Pesquisas Espaciais)	1859-1975
14	HadGEM2-CC		1859-1975
15	HadGEM2-ES		1859-1975
16	IPSL-CM5A-MR	Institute Pierre-Simon Laplace	1850-1975
17	MIROC5	Atmosphere and Ocean Research Institute (The University of Tokyo), National Institute for Environmental Studies, and Japan Agency for Marine-Earth Science and Technology	1850-1975
18	MPI-ESM-LR	Max-Planck-Institut fur Meteorologie (Max Planck Institute for Meteorology)	1850-1975
19	MRI-CGCM3	Meteorological Research Institute	1850-1975
20	NORESM1-M	Norwegian Climate Centre	1850-1975
21	NORESM1-ME		1850-1975

$$v_{i,j}^{x,y,z} = \tanh\left(\sum_{m=1}^{M_{i-1}}\sum_{p=1}^{P_i}\sum_{q=1}^{Q_i}\sum_{r=1}^{R_i} w_{i,j,\mathrm{m}}^{p,q,r} v_{(i-1),m}^{(x+p-P_i/2,y+q-Q_i/2,z+r-R_i/2)} + b_{i,j}\right) \quad (1)$$

In equation (1) $v_{i,j}^{x,y,z}$ is the value of the *j*th feature map in the *i*th convolutional layer at grid point (x,y,z) and P_i and Q_i stand for the longitudinal and transverse coordinates of the

convolutional filter for this layer, respectively. The tanh function is chosen as an activation function to consider the symmetric nature of El Niño and La Niña. The dimensions of the convolutional filter are set to $8 \times 4 \times 2$ during the first convolutional process (that is, P_i= 8; Q_i= 4; R_i= 2), and to $4 \times 2 \times 2$ during the second and third convolutional processes. The number of feature maps in the *(i−1)*th layer is determined by M_{i-1} and $w_{i,j,\mathrm{m}}^{p,q,r}$ stand for the weight at grid point (p,q,r) in the 3D convolutional filter. $v_{(i-1),\,m}^{(x+p-P_i/2,y+q-Q_i/2,z+r-R_i/2)}$ present the value of the mth feature map for the $(i-1)$th convolutional layer at the grid point $(x+p-P_i/2\,,y+q-Q_i/2\,,z+r-R_i/2)$. It should be noted that the number of channels is one in this study, as the input tensors are not RGB images. Finally, $b_{i,j}$ represent the bias of the *j*th feature map in the *i*th convolutional layer.

The 3DCNN-based statistical model for ENSO forecasting receives a $[l_1,l_2,l_3,l_4]$ input tensor, then extracts its features by multiplying K number of $\frac{l_1}{9} \times \frac{l_2}{6} \times \frac{l_3}{2} \times \frac{l_4}{1}$ 3D kernels and follows with a 3D average pooling operation with kernel size of $Ag_1 \times Ag_2 \times Ag_3$. In this study, input maps has only one channel; so, the first convolution layer gets the size of $\frac{l_1}{Ag_1} \times \frac{l_2}{Ag_2} \times \frac{l_3}{Ag_3} \times 1$ that multiplies to the next 3D convolutional kernel with the size of $\frac{l_1}{18} \times \frac{l_2}{12} \times 1 \times K$. Then, a 3D max-pooling operator with the size of $Mx_1 \times Mx_2 \times Mx_3$, in which $Mx_1 = Mx_2 = 2$ and $Mx_3 = 1$, reduces the input tensor to a $\frac{l_1}{Ag_1 \cdot Mx_1} \times \frac{l_2}{Ag_2 \cdot Mx_2} \times \frac{l_3}{Ag_3 \cdot Mx_3} \times 1$ tensor. Then, another 3D convolution kernel with the dimension of previous kernel implies and delivers the results to a reshape operator that transforms the tensor to a vector with the dimension $\left[1, K \times \frac{l_1}{Ag_1 \cdot Mx_1} \times \frac{l_2}{Ag_2 \cdot Mx_2} \times \frac{l_3}{Ag_3 \cdot Mx_3}\right]$ that multiplies again with a $\left[K \times \frac{l_1}{Ag_1 \cdot Mx_1} \times \frac{l_2}{Ag_2 \cdot Mx_2} \times \frac{l_3}{Ag_3 \cdot Mx_3}, F\right]$ vector. The resultant is a $1 \times F$ vector that connects to the single Nino 3.4 index after a dot product by the last weight vector of $F \times 1$.

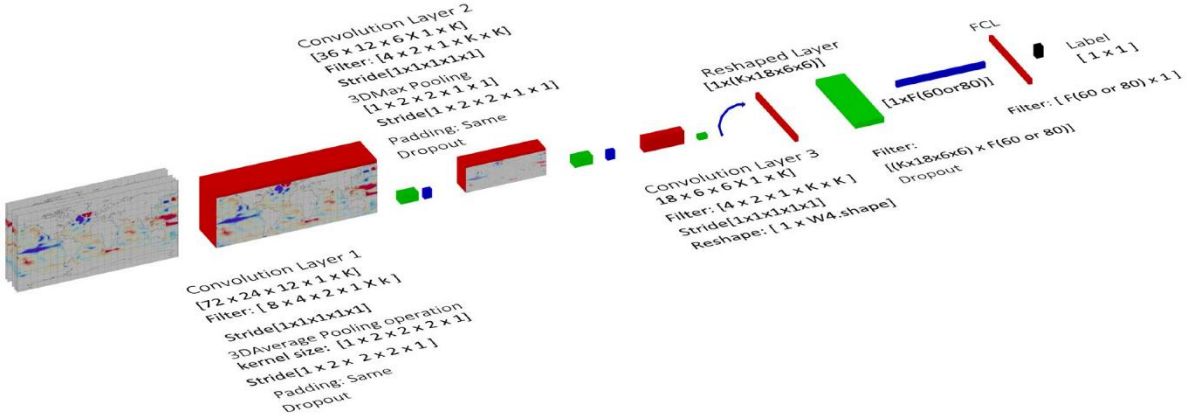


Figure 1. 3DCNN Network Architecture and input maps

Three 3D convolutional layers, one 3D average pooling operator, and one 3D max-pooling layer make up the 3DCNN module. To extract temporal dependencies from the input streams, we utilize a 3D average pooling. The max-pooling operator selects the maximum value from each 2 × 2 grid, and the last convolutional layer links to the one-by-one scalar label through a dense layer. The 3DCNN module is formulated separately for each forecast lead month and target season. We set the number of convolutional filters to either 5 or 7, and the number of neurons in the fully connected layer is either 50 or 70. The model runs up to 1200 times to reach a stable result. We enhanced the prediction by analyzing the structural similarity of multiple 3DCNN outputs using a GNN-based representation framework. This ensemble procedure systematically increases the accuracy of the module. The ROA Method (Talaat & Gamel, 2022) determines the optimum size of the mini-batch for each lead and season combination. The learning rate is set to 0.005, dropout is used, and the Xavier initialization is applied to define the initial weights.

2.3. Time Series Informer

At first glance, one might attempt to capture the long-term temporal evolution of ENSO within a deep learning framework by supplying the model with an extended sequence of anomaly images. However, this approach leads to an exponential increase in the number of parameters, rendering accurate prediction feasible only for short-term forecasts. In addition, such lengthy input sequences introduce a significant amount of irrelevant information, which can distract the model and degrade its performance. Conversely, using only a few concatenated maps fails to provide sufficient temporal context compared to a full time series. Consequently, a practical strategy for incorporating long-term temporal features is to analyze the temporal dynamics of ENSO separately. In this study, we employ the ENSO time series as the input to the Time Series Informer (TSI) module to forecast its temporal evolution. The results from this model are then integrated through an ensemble approach to produce a comprehensive multimodal prediction.

Transformer neural networks (Vaswani, Shazeer, & Parmar, 2017; Castangia, Grajales, Aliberti, & Rossi, 2023) are machine learning models proposed by Vaswani et al. mainly to overcome LSTM drawbacks (Ho & Ermon, 2016). The concept of this method is the attention not only to the events that occur before a specific predictand, but also to the events that happen after it. This concept has attracted much attention from researchers and scientists in many research areas, such as information retrieval, text classification, and document summarization. In addition to language-related applications, Transformers have also been utilized in computer vision (Luc et al., 2016), chemistry (Schwaller et al., 2019), life sciences (Rives et al., 2016), and river level prediction (Gao & Zhang, 2017; Nicolas et al., 2020**).** ENSO forecasting is one of the latest applications of Transformers, recently done by Ye et al. (Ye et al., 2022). They have asserted that the model can predict ENSO even better than CNN (Ham, Kim, & Luo, 2019) one and a half years ahead; however, the model still suffers from some drawbacks. For

example, the skill of the method for strong ENSO events is relatively low. Furthermore, only for the spring season, the predictions are more accurate than those of other state-of-the-art models.

Inspired by the work of Zhou et al. (Zhou et al., 2021), we feed long-term ENSO time series into the Informer network and receive a two-year-long prediction. The overall architecture of the Informer network $\mathcal{G}_{\theta_G}$, as shown in Figure 2, consists of five main components (Hassani, Javadi, & Naisipour, 2025).

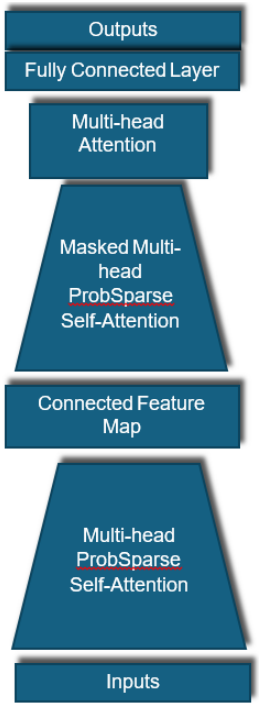


Figure 2 . Overall view of Informer Module.

Given a time sequence $\mathcal{X}^t = \{x_1^t,\ldots,x_{L_x}^t \mid x_i^t \in \mathbb{R}^{dx}\}$ at time t and the output corresponding sequence $\mathcal{Y}^t = \{y_1^t,\ldots,y_{L_x}^t \mid y_i^t \in \mathbb{R}^{dy}\}$, the module encodes the input representations $\mathcal{X}^t$ into a hidden state representation $\mathcal{H}^t$ and decodes an output representations $\mathcal{Y}^t$ from $\mathcal{H}^t = \{h_1^t,\ldots,h_{L_h}^t\}$. The extracted features of each series are added to the positional encoders, after the input series have been embedded and representative features have been created by the embedding module. We consider that we have t-th sequence input $\mathcal{X}^t$ and p types of global time stamps and the feature dimension after input representation is d_{model}. We first use a fixed position embedding to maintain the local context:

$$\mathbf{PE}_{(pos,2j)} = \sin\left(pos/(2L_x)^{2j/d_{model}}\right) \quad (2)$$

$$\mathbf{PE}_{(pos,2j+1)} = \cos\left(pos/(2L_x)^{2j/d_{model}}\right) \quad (3)$$

Where $\boldsymbol{j} \in \{\mathbf{1}, \ldots, \boldsymbol{d_{model}/2}\}$. A learnable stamp embedding $\mathbf{SE}_{(\boldsymbol{pos})}$ utilizes every global time stamp with limited vocab size. Since the self-attention's similarity computation can have access to global context, the computation consuming is affordable on long inputs. we project the scalar context $\mathbf{x}_i^t$ into $\boldsymbol{d_{model}}$-dim vector $\mathbf{u}_i^t$ with 1-D convolutional filters with kernel width equal to 3 and stride one to align the dimension. Thus, we have the feeding vector:

$$\boldsymbol{\mathcal{X}}_{\boldsymbol{feed}[i]}^{\boldsymbol{t}} = \boldsymbol{\alpha}\mathbf{u}_i^t + \mathbf{PE}_{(\boldsymbol{L_x}\times(\boldsymbol{t}-\mathbf{1})+\boldsymbol{i},)} + \sum_{\boldsymbol{p}}\left[\mathbf{SE}_{(\boldsymbol{L_x}\times(\boldsymbol{t}-\mathbf{1})+\boldsymbol{i})}\right]_{\boldsymbol{p}} \quad (4)$$

In this equation, $\boldsymbol{\alpha}$ is the factor balancing the magnitude between the scalar projection and local/global embedding that is set to 1 in this study, and $\boldsymbol{i} \in \{\mathbf{1}, \ldots, \boldsymbol{L_x}\}$.

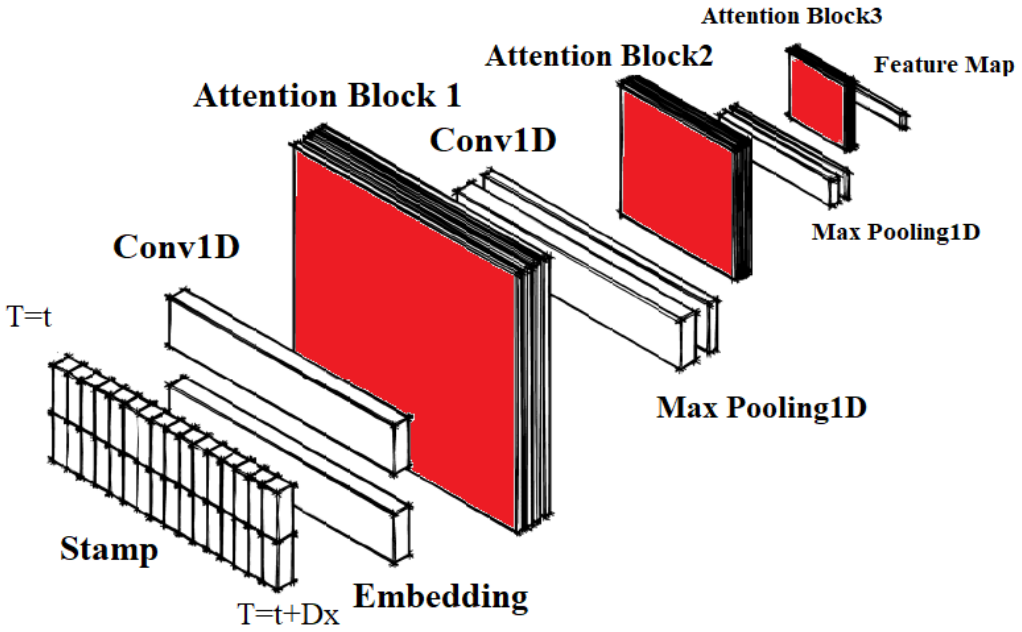


Figure 3. The encoder's single stack in the Informer module.

ProbSparse Self-attention. We apply the ProbSparse self-attention by allowing each key to only attend to the u dominant queries:

$$\mathcal{A}(\mathrm{Q},\mathrm{K},\mathrm{V}) = \mathrm{Softmax}\left(\frac{\bar{\mathrm{Q}}\mathrm{K}^{\mathrm{T}}}{\sqrt{d}}\right)V \quad (5)$$

Where Q is a sparse matrix of the same size of q and it only contains the Top-u queries under the sparsity measurement $M(\mathrm{q},\mathrm{K})$.

Encoder. The encoder's purpose is to extract from the lengthy sequential inputs the reliable long-range dependency. As the input representation is done, the matrix $\mathrm{X}_{en}^{t} \in \mathbb{R}^{L_x \times d_{model}}$ will be the new shape of the *t*-th sequence input $\boldsymbol{\mathcal{X}^t}$.

Self-attention Distilling. We use the distilling operation to privilege the superior ones with dominating features, because the natural consequence of the ProbSparse self-attention mechanism has redundant combinations of value **V** (Reiser, Neubert, & Eberhard, 2022). This also makes a focused self-attention feature map in the next layer. Observing the n-heads weights matrix of the Attention blocks in Figure 3, it drastically reduces the input's time dimension. According to equation (6), the applied procedure forwards from the *j*-th layer into the (*j* + 1)-th layer.

$$X_{j+1}^{t} = \text{MaxPool}\left(\text{ELU}\left(\text{Conv1d}\left(\left[X_{j}^{t}\right]_{AB}\right)\right)\right) \tag{6}$$

In this formula, $[\cdot]_{AB}$ represents the attention block, ELU(·) is the activation function and Conv1d(·) is a 1D convolutional filter with a kernel width of three. To reduce the whole memory usage, a max-pooling layer with a stride 2 down-samples X^t into its half slice.

Decoder. We use a stack of two identical multi-head attention layers as the decoder, and to alleviate the speed plunge in long prediction, we apply the generative inference. The following vectors are given to the vanilla decoder as the input:

$$X_{de}^{t} = \text{Concat}\left(X_{token}^{t}, X_{0}^{t}\right) \in \mathbb{R}^{\left(L_{token}+L_{y}\right)\times d_{model}} \tag{7}$$

where $X_{token}^{t} \in \mathbb{R}^{L_{token}\times d_{model}}$ and $X_{0}^{t} \in \mathbb{R}^{L_{y}\times d_{model}}$ are is the start token and placeholder for the target sequence, respectively. The final layer is a dense layer that generates the results.

Generative Inference. We use one of the best data augmentation methods called Generative Adversarial Network (GAN) to increase the input data that can be trained to imitate a data distribution. GAN was first proposed in 2014 by Ian Goodfellow et al. (Goodfellow et al., 2004). This method learns to produce fresh data with the same statistics as the training set, given a training set. GANs have been shown to be beneficial for semi-supervised learning (Salimans et al., 2016) while being first proposed as a type of generative model for unsupervised learning. They have also been successful in fully supervised learning (Isola et al., 2017) and reinforcement learning (Ho & Ermon, 2016). GANs can improve astronomical images (Schawinski et al., 2017) and simulate gravitational lensing for dark matter research (Mustafa et al., 2019). They were used to model the distribution of dark matter in the universe and to predict the occurrence of gravitational lensing (Paganini, de Oliveira, & Nachman, 2017). GANs have been proposed as a fast and accurate way of modeling high-energy jet formation (Erdmann, Glombitza, & Quast, 2019) and modeling showers through calorimeters of high-energy physics experiments (Salamani et al., 2018). They have also been trained to approximate bottlenecks in simulations of particle physics experiments (Luc et al., 2016).

The GAN consists of "indirect" training through the discriminator and another neural network that can tell how "realistic" the input seems, which itself is also being updated dynamically.

This means that the generator is not trained to minimize the distance to a specific image, but rather to fool the discriminator. In this study, each probability space (Ω, μ_{ref}) defines a GAN game. There are two players called the generator and discriminator. The generator's strategy set is P (Ω), the set of all probability measures μ_G on (Ω). The discriminator's strategy set is the set of Markov kernels μ_D: $\Omega \rightarrow$ P [0,1], where P[0,1] is the set of probability measures on [0,1]. The GAN game is a zero-sum game, with the objective function:

$$L(\mu_G,\mu_D) \coloneqq \mathbb{E}_{x\sim y_{ref},y\sim\mu_D(x)}[\ln y] + \mathbb{E}_{x\sim y_G,y\sim\mu_D(x)}[\ln(1-y)] \quad (8)$$

The generator aims to minimize the objective, and the discriminator aims to maximize the objective. The generative network's training goal is to increase the error rate of the discriminative network (Mohamed & Lakshminarayanan, 2016; Andrej, 2016). GANs are implicit generative models (Blöschl, Hall, & Viglione, 2019) that do not explicitly model the likelihood function nor provide a means for finding the latent variable corresponding to a given sample.

Start token is efficiently applied in NLP's "dynamic decoding" (Mohamed & Lakshminarayanan, 2016), and we extend it into a generative way. We sample a L_{token} long sequence in the input sequence, so X_0 contains the target sequence's time stamp, that is, the context at the target. Instead of the cumbersome "dynamic decoding" in the vanilla encoder-decoder blocks, the decoder used in this study forecasts outputs by a forward procedure. We chose the MSE loss function across the entire Informer module. Together with the 3DCNN module, the TSI forms the dual-branch foundation of the MEF model, generating 40 ensemble members each for subsequent graph-based integration.

2.4. Preprocessing & Training Configuration

All predictor variables, including sea surface temperature (SST) and upper 300-m heat content (HC300), were first converted into monthly anomalies with respect to the 1981–2010 climatology. The variables were regridded onto a common 5° × 5° latitude–longitude grid by applying bilinear regridding for the purposes of ensuring spatial consistency between sources. Temporal alignment was then applied for synchronizing CMIP-based predictors and observational targets. Missing or spurious values were linearly filled to maintain the continuity of each monthly series. Before training, all inputs were standardized to have zero mean and unit variance. The system's predictive module is a 3D-Convolutional Neural Network (3DCNN), trained for predicting Niño-3.4 index anomalies. It was trained with the Adam optimizer, learning rate 0.005, Xavier (Glorot) weight initialization, and dropout = 0.3 as a measure against overfitting. The loss function is the mean-squared error (MSE) between predicted and observed anomalies. Training was carried out for up to 1200 stochastic initializations; 40 of these runs per lead–month group were retained to form the ensemble for the next stage.

The network utilized mini-batches grouped by lead and calendar month, and early stopping with 15-epoch patience was employed to prevent degradation of validation skill. All training and validation were conducted independently for each lead–month pair to preserve seasonal dependence. The ensuing 40 runs for each group are the candidate ensemble members that are subsequently validated and selected by the next graph-based model.

2.5. GNN-based Structural Similarity Analysis

The emergence of Graph Neural Networks (GNNs) (Taccari, Wang, & Nuttall, 2024; Wu, 2025; Ali, Richter, & Ertürk, 2025) marked a pivotal shift in machine learning, extending neural architectures to non-Euclidean, graph-structured data. GNNs are generalized recurrent neural networks to graphs, using iterative message passing to update node representations based on their neighbors' features. This addressed the limitations of traditional neural networks, which struggled with relational data in domains like social networks (Cai, McPhaden, & Grimm, 2020), chemistry (L'Heureux et al., 2020), and climate science (Moazezi Khah Tehran et al., 2025). GNNs have emerged as a transformative approach for modeling complex, non-Euclidean data structures, making them particularly suited for climate science applications where data often exhibit relational, spatiotemporal dependencies. The proposed study, which leverages GNNs for optimizing ensemble predictions in El Niño–Southern Oscillation (ENSO) forecasting by constructing similarity graphs among ensemble members, aligns with a growing body of work applying GNNs to climate-related challenges.

The application of GNNs to climate science builds on their ability to model relational data, where nodes represent entities (e.g., geographic locations, climate variables) and edges encode interactions (e.g., teleconnections, physical processes).

A graph is defined as:

- $G = (V, E)$ where:
  - $V = \{v_1, v_2, \ldots, v_n\}$ is the set of nodes.
  - $E \subseteq V \times V$ is the set of edges.
- $A \in \mathbb{R}^{n \times n}$ is the adjacency matrix, where $A_{ij} = 1$ if there is an edge between nodes v_i and v_j.
- Each node v_i has a feature vector $x_i \in \mathbb{R}^d$.

The goal is to learn node embeddings $\mathbf{h}_i^L$ after L layers that encode both features and structure.

Most GNNs follow the **Message Passing Neural Network (MPNN)** framework:

1. **Message Aggregation:**
   $m_i^{l+1} = \text{AGGREGATE}^l(\{ \mathbf{h}_j^l : j \in N(i) \})$

where N(i) is the set of neighbors of node i.

2. **Node Update:**
   $\mathbf{h}_i^{l+1} = \text{UPDATE}^l(\mathbf{h}_i^l, m_i^{l+1})$

Here $\mathbf{h}_i^0 = \mathbf{x}_i$ (initial node features). After L layers, $\mathbf{h}_i^L$ is used for tasks like classification or link prediction.

GNNs are theoretically linked to the **Weisfeiler–Lehman (WL) graph isomorphism test**, which iteratively updates node labels based on neighbors. Standard GNNs have the same expressive power as the 1-WL test, meaning they cannot distinguish all non-isomorphic graphs. More advanced variants (e.g., Graph Isomorphism Networks, higher-order GNNs) aim to improve this.

The loss function depends on the task:

- **Node classification:**
- $L = -\sum\sum y_{ic} \log(\hat{y}_{ic})$
- **Graph classification:**
  Use READOUT(H^L) to pool node embeddings, then apply classification loss.
- **Link prediction:**
  Use binary cross-entropy on predicted edge probabilities.

GNN-based Structural Similarity Analysis

In an attempt to transcend the limitations of modeling long-term temporal dependencies using traditional architectures, we propose a different solution grounded in structural representations of graphs. Rather than relying solely on sequence models, our approach examines similarity in outputs among members of the MEF ensemble (3DCNN + TSI) using graph-based approaches. Every run of the model (of the 80 members of the MEF ensemble) is a node in a weighted undirected graph whose weight across every edge represents pairwise similarity in the forecasts (e.g., RMSE or correlation). The interactions allow us to model ensemble space as a graph where topological structure captures behavioral similarity.

By incorporating node-level data (e.g., error statistics, temporal smoothness), a Graph Neural Network (GNN) is trained to uncover the latent relationship among model runs (Labibzadeh et al., 2015; Naisipour, Afshar, Hassani, & Zeinali, 2009). This allows for improved clustering, filtering, or voting approaches to identify representative or resilient predictions, particularly at longer lead times where traditional methods might degrade.

Diversity of models in ensemble deep learning systems, especially for climate prediction tasks such as ENSO prediction, is a double-edged sword. While repeated runs of the model may

provide robust uncertainty estimates, selecting the most representative and reliable forecasts from amongst them is not a trivial task. In this paper, we propose a theoretical approach to investigating the internal structure of ensemble outputs using Graph Neural Networks (GNNs).

GNNs are a kind of neural network that is particularly designed to process data that can be represented in the form of graphs, ordered sets of nodes and edges, where information is relational, not purely sequential or spatial. GNNs, as opposed to standard neural networks, can capture interdependencies between entities that are not sequentially or spatially related. This qualifies them to represent similarities between independent deep model runs.

In our proposed framework, all 80 model runs generated by the 3DCNN are treated as nodes in a graph. Edges between these nodes represent the degree of similarity between pairs of runs. Similarity can be determined on the basis of a number of metrics, such as:

- Root Mean Square Error (RMSE): absolute error of prediction with respect to the ground truth.
- Pearson Correlation Coefficient: linear consistency in temporal dynamics
- Cosine Similarity: capturing directional alignment in multivariate time-series space
- Dynamic Time Warping (DTW): allowing for flexible alignment of sequences to account for temporal phase shifts.

The resulting structure is an undirected, weighted graph $G(V, E)$ where each node $V \ni v_i$ represents an individual model run, and each edge $E \ni e_{ij}$ has a weight corresponding to the similarity between runs $i \; and \; j$. Optional node features can be added to each V_i , such as error metrics, temporal smoothness metrics, or entropy-based descriptors. Subsequently, one may use a GNN to learn an embedding for each run by aggregating information from its neighbors. The unsupervised representation captures not only the solo performance of each run but also its position within the global ensemble structure. One can use these embeddings for:

- Clustering similar runs into consistent clusters
- Detection of anomalous or inconsistent outputs
- Improving ensemble strategies by weighting runs based on structural stability

Importantly, the GNN is not used here as a forecasting model, but rather as a meta-modeling tool, a means of interpreting the ensemble space. This approach allows us to move beyond scalar error metrics and gain a relational understanding of model behavior (Reiser, Neubert, & Eberhard, 2022), helping to uncover patterns of agreement, divergence, or redundancy within ensemble outputs. This graph-theoretic approach offers new possibilities for structured ensemble analysis in climate modeling and is relevant to a wide range of multimodal or multi-run systems beyond ENSO prediction.

2.6. Graph Construction & Supervised Selection

After computing the similarity metrics among ensemble members, the resulting matrices are transformed into graph structures suitable for supervised learning. In order to consolidate the ensemble forecasts produced by the MEF model, we represent the collection of runs for a single lead–month pair as a graph representing statistical similarity among members. Each graph consists of 80 nodes, representing the MEF ensemble members (40 from the 3DCNN module and 40 from the TSI module). Edges between nodes quantify pairwise similarity in the temporal behavior of the forecast Niño-3.4 index, computed with Pearson correlation and rescaled to the range [0, 1]. Such a representation allows the model to use relational knowledge and identify groups of runs with consistent dynamics. The graph model is supervised-trained.

For every set of leads–months, the run with the most out-of-sample correlation and lowest RMSE with observations is labeled as the positive (target) node, and the remaining 39 nodes are labeled as negative. For the 23 lead times and 12 calendar months (276 graphs total), we divide the graphs at random into the training set 75 % and the remaining 25 % to test.

This supervised setting teaches the model to recognize underlying statistical patterns that characterize high-skill ensemble members rather than clustering them in an unsupervised way.

Input features and learning objective: For each node, a compact feature representation is derived from descriptive statistics of the corresponding time series, e.g., mean, variance, lag-1 autocorrelation, spectral energy in selected frequency bands, and pairwise similarity scores. These are normalized and appended to the adjacency matrix to form the graph input.

The Graph Convolutional Network (GCN) calculates the node features by successively propagating messages through adjacent layers, enabling every node to gather information on neighboring nodes. The network makes a prediction of probability distribution over the best-performing run likelihood nodes.

Loss function is categorical cross-entropy between output probabilities and ground truth binary labels. Optimization is done using Adam with learning rate = 0.005 and dropout = 0.3, along with early stopping based on validation loss.

Interpretation and behavior: Although the GCN receives no direct correspondence or RMSE values as input, it learns to do so implicitly and identifies time-series patterns that correspond to high predictive capacity, e.g., coherent large-scale oscillations, moderate variance, and low high-frequency noise. In inference on unseen lead–month graphs, the model puts maximum probability on the node whose statistical structure most closely matches those of previously observed high-skill members.

It accomplishes this by mechanizing the selection of ensemble members: rather than taking a plain mean of all 80 members, the GNN picks the best estimate for every lead time, enhancing stability as well as long-lead accuracy.

2.7. GNN Architecture & Regularization

The proposed graph network is achieved in the form of a three-layer Graph Convolutional Network (GCN) that propagates relational knowledge among ensemble-member nodes to find the most competent prediction.

Each of the graphs is for one lead–month group with 80 nodes (the ensemble members of the MEF model) and an adjacency matrix that encodes their pairwise temporal similarity.

By the message-passing procedure, each node sequentially combines information from its neighbors so that the network learns higher-order structural features characterizing inter-member coherence.

Mathematical formulation: The layer-wise propagation rule follows the standard GCN formulation:

$$\boldsymbol{H^{(l+1)} = \sigma\left(\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right)} \tag{9}$$

where $\boldsymbol{H^{(l)}}$and $\boldsymbol{H^{(l+1)}}$denote the node-feature matrices at layers $\boldsymbol{l}$and $\boldsymbol{l+1}$; $\boldsymbol{W^{(l)}}$retrainable weight matrices; $\boldsymbol{\widetilde{A} = A + I}$ is the adjacency matrix with self-connections; $\boldsymbol{\widetilde{D}}$ is the corresponding degree matrix; and $\boldsymbol{\sigma(\cdot)}$ is a nonlinear activation function (ReLU). This operation smooths features over neighboring nodes and extracts latent structural correlations that traditional averaging techniques cannot capture.

Table 3. Network configuration and hyperparameter settings used in the CNN–GNN hybrid model for ENSO prediction

Component	Specification
Input	80 nodes per graph (the MEF ensemble: 40 from 3DCNN + 40 from TSI), each with 5–10 statistical features (mean, variance, lag-1 autocorrelation, spectral energy, similarity indices, etc.)
Hidden layers	2 graph-convolution layers, each with 64 hidden units
Activation	Rectified Linear Unit (ReLU)
Output	1 fully connected classification head producing node-level probabilities of being the top-performing run
Loss function	Categorical cross-entropy between predicted probabilities and binary labels
Optimizer	Adam (learning rate = 0.005, $\beta_1 = 0.9$, $\beta_2 = 0.999$)
Dropout	0.3 between layers
Epochs/patience	≤ 200 epochs with early stopping (patience = 10)
Batch size	1 graph per mini-batch

All the calculations were performed in PyTorch Geometric using GPU acceleration (NVIDIA A100). The weights were initialized with Xavier (Glorot) initialization, and training used cross-entropy loss to encourage positive and negative node class separation.
Regularization and generalization: Overfitting control was a priority because there were few graphs available (276 lead–month combinations).
Three strategies that complemented each other were used:

1) Data partitioning. A 75 %/25 % train–test split ensured validation graphs were from unseen lead–month pairs to prevent leakage.
2) Dropout. Randomly masking 30 % of neuron activations per layer during training forced the network to learn redundant, tolerant representations.
3) Early stopping. Training was automatically halted once validation loss no longer decreased for 10 epochs, which prevented unnecessary weight adaptation.
These together effectively prevented memorization of individual graphs and fostered generalization across regimes of ENSO. In evaluation, the model consistently yielded the same top-performing node even when small perturbations to the input graph were introduced, citing architectural stability.

Behavior and interpretation of model: GCN does not receive direct correlation or RMSE values but instead learns its own internal notion of "skillfulness" from relational statistics in the graph. Nodes corresponding to runs with smoother temporal organization and consistent large-scale variance tend to have more activation responses in the final layer.
Therefore, the learned representation can be interpreted as an implicit mapping between structural similarity and predictive consistency. Although the network occasionally gets borderline cases wrong, the aggregate ensemble chosen by the GNN is more stable at long leads and less variable in error than uniform averaging. Subsequent research can extend this architecture with attention-based variants such as Graph Attention Networks (GAT) or GraphSAGE to address asymmetric interdependencies between ensemble members and improve interpretability.

2.8. Validation Period and Rationale

The 2000–2017 validation window was chosen specifically in order to test the proposed framework over an era of extreme ENSO variability and nonlinear tropical Pacific climate behavior. In contrast to the late 20th century, the early 21st century has been marked by increased event asymmetry, repeated toggling between El Niño and La Niña phases, and increased interaction between the oceanic thermocline and atmospheric Walker circulation. These features render this era a science-challenging proving ground for any data-limited ENSO prediction system.

This 18-year record holds several historically significant events, such as the 2002–2003, 2009–2010, and 2015–2016 El Niño events, as well as several La Niña events (e.g., 2007–2008 and

2010–2011). It also captures repeated occurrences of the spring predictability barrier, recently witnessed as one of the most trying times in ENSO onset and demise prediction. Model testing across these regimes enables a rigorous assessment of its ability to continue to be accurate under rapidly evolving coupled-ocean–atmosphere dynamics. Furthermore, the specific interval has been considerably addressed in recent literature as a climatically transitional interval, marked with fluctuations in mean-state warming, Pacific decadal variability, and amplitude modulation of ENSO (Ham, Kim, & Luo, 2019).

With the focus placed on this complex interval, the proposed GNN-strengthened MEF scheme is compelled to explain the stationary as well as non-stationary nature of the ENSO system. The relatively short but highly dynamic time scale allows for more concentrated evaluation of model generalization, as it covers multiple ENSO regimes over fewer years and therefore highlights the model's flexibility rather than its ability to replicate long climatic means. A second reason to select this window is the presence and consistency of good-quality observational data from the GODAS reanalysis system, following greater reliability after the late 1990s due to satellite-era assimilation and improved ocean subsurface observations.

This ensures that validation results are free from observational uncertainty and provide a robust standard for comparison of forecasted and observed Niño-3.4 anomalies. Although more extended historic records exist, the 2000–2017 epoch is the best balance of event richness, computational manageability, and data quality. Testing the model on this epoch, therefore, provides a focused but challenging benchmark that is particularly appropriate for highlighting the predictive power, stability, and physical interpretability of the new graph-based ensemble-selection approach.

2.9. CMIP5 Data Usage and Bias Correction Clarification

To make clear the function of CMIP5 data to avoid confusion, it is required to define exactly how such simulations were put into practice under the proposed hybrid framework. The CMIP5 datasets were applied only in the 3DCNN training step to provide a broad ensemble of spatiotemporal samples among various global climate models. The procedure allowed the convolutional network to learn generalized spatial–temporal representations of oceanic variability as the candidate forecasts. All forty of the CNN ensemble outputs per lead–month combination were then taken individually to be a predictive member and used as input to the graph-based ensemble-selection module. The Graph Neural Network (GNN) step therefore only deals with the MEF-produced time-series outputs and never reads or combines CMIP5 predictor variables with observational fields directly. No explicit bias correction or domain adaptation process is therefore ever required in the GNN pipeline. All of the consistency fixes required were already addressed in the preprocessing and training configuration described in Section 2.2, including:

- Spatial reharmonization of predictor grids to a uniform 5° × 5° resolution through bilinear regridding.
- Normalization of anomalies versus a 1981–2010 climatology to remove mean offsets between models.
- Temporal synchronization of all fields to common monthly time stamps.

These phases ensure that all CNN inputs—and thus their outputs—share a standard statistical foundation prior to the graph-learning process.

This modularity, as a design principle, keeps systematic model bias localized in the CNN module and limits its propagation to the GNN module.

The GNN learns only about the relative statistical and structural variations among ensemble members, irrespective of the particular CMIP5 model's bias character. This separation of learning responsibilities provides two advantages:

(i) it minimizes potential graph model pollution from inherited simulation biases, and

(ii) it increases interpretability since choices by the GNN are made purely on the basis of data-driven relational properties rather than on the basis of explicit corrections for climate-model mistakes.

Moreover, not including bias-correction and domain-adaptation steps at the GNN level aligns with our conceptual goal: constructing a post-processing selection framework that selects the most dynamically consistent ensemble member out of the given predictions, but not recalibrating or even altering the underlying physics of the CMIP5 simulations. This approach provides a cleaner separation of forecast generation (CNN) and forecast selection (GNN), therefore rendering the whole system more transparent, scalable, and reproducible.

2.10. Implementation of Code and Cloud Super Computing

All experiments were implemented in Python using the TensorFlow framework, executed on Google Colab’s GPU-enabled environment (NVIDIA A100). Due to the large volume of data and complex computer calculations, we used the computing capabilities of the Google Colabatory framework.

Ultimately, the GNN-based ensemble selection mechanism automatically identifies the most reliable predictions among all MEF ensemble members, yielding a final optimized forecast. Uncertainty analysis is conducted to ascertain the voting values for each module’s ensemble members; specifically, greater uncertainty associated with a module results in a correspondingly lower voting value. The final ensemble prediction is computed as the weighted average of the individual results. Figure 4 illustrates the architecture of the comprehensive forecasting framework. As previously noted, the MEF model comprises three key components: 3DCNN, Informers, and the GNN fusion modules.

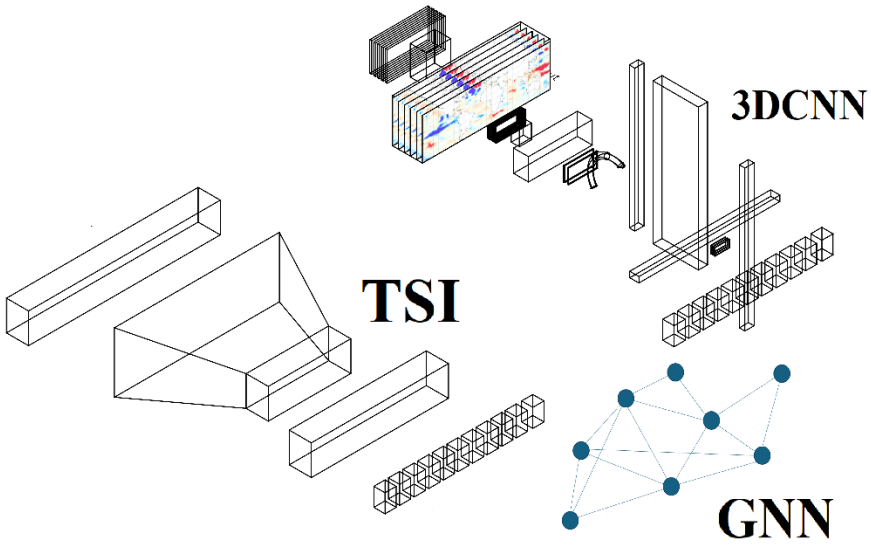


Figure 4. the procedure of the complete forecasting package.

3. Results and Discussion

In this study, we developed a method for accurately forecasting the ENSO for a duration of up to two years. Given that the variability of ENSO has increased over the past two decades, it is more appropriate to assess model skillfulness during the post-2000 period (Ham, Kim, & Luo, 2019; Cai, McPhaden, & Grimm, 2020; Wang, Cheng, Zhang, & Yu, 2023). To further complement the stability and interpretability of our forecasting system, we suggested a graph-based approach to investigate the structural similarity between the 80 runs of the ensemble model. In this approach, every run is labeled as a node in an undirected weighted graph. The nodes are connected with edges that are labeled based on pairwise similarity measures. This architecture allows us to overcome scalar performance metrics by the capacity to observe behavioral signatures and temporal consistency between outputs. Runs exhibiting similar forecasting behaviors produce clusters, indicating cohesion between best-performing models. Single nodes, however, may suggest outliers or unstable forecasts. Using community detection or centrality analysis, the majority of structurally reliable runs can be identified. A graph-based approach like this provides noise filtering on a data-driven basis, more effective ensemble aggregation, and a clearer view of the performance of the model across different lead times and seasonal conditions. Especially when one is dealing with long-range forecasts, where uncertainty is higher, the method enables the selection of forecasts that are not just accurate but also robust within the ensemble space. Lastly, this structural perspective bridges the performance score-based assessment process to a network-influenced selection strategy, paving the way for more interpretable and more robust climate prediction models.

3.1. Results Evaluation

We developed the enhanced MEF model specific to post-2000 ENSO prediction. In evaluating

MEF, trend-sensitive (e.g., correlation coefficient) and error-based (e.g., RMSE) metrics were applied, giving a well-rounded performance comparison. For shorter lead times (1–3 months), the original MEF has a slightly smaller RMSE than the enhanced model. However, always scores higher in correlation values. With an increasing forecast horizon, EMEF begins to dominate MEF significantly. From 5 to 14-month lead times, EMEF performs superiorly in terms of predictability.

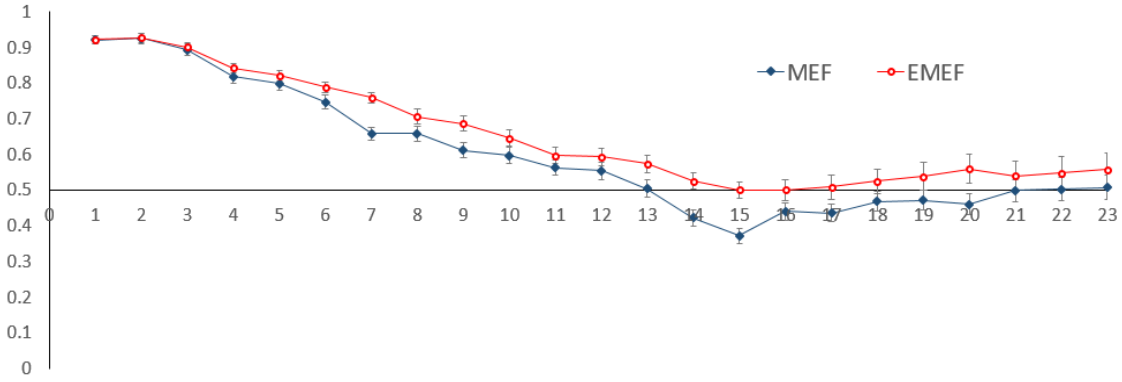


Figure 5. Comparison between the GNN-based and MEF ensemble-selection methods in correlation skill for different lead times (1–23 months). The x-axis represents lead time (months), and the y-axis is the correlation between the predicted and observed Niño-3.4 index. The new GNN-based method performs systematically better than MEF by approximately 10 % across all lead times, particularly for longer lead times, thanks to the effective identification and merging of top-performing ensemble members based on graph-structural similarity.

We conducted a more structured analysis through an ensemble output graph-based comparison. In that case, each 80 ensemble runs constitute a node in the similarity graph whose edge weights represent closeness in RMSE or correlation. Clustering structures in the graph capture clusters of trustworthy high-performing runs, and outlier nodes reveal unstable behavior. By analyzing this structure, MEF allows selection based on the best bets, not just the highest scoring ones individually, but also the ones whose structures best resemble other top contenders. This graph-based ensemble filter enables a stronger forecasting process, especially for longer lead times where the uncertainty is higher. EMEF's structural integration and ensemble optimization strategies enable its capability to counter the novel challenges of post-2000 ENSO forecasting more effectively than traditional deep learning approaches.

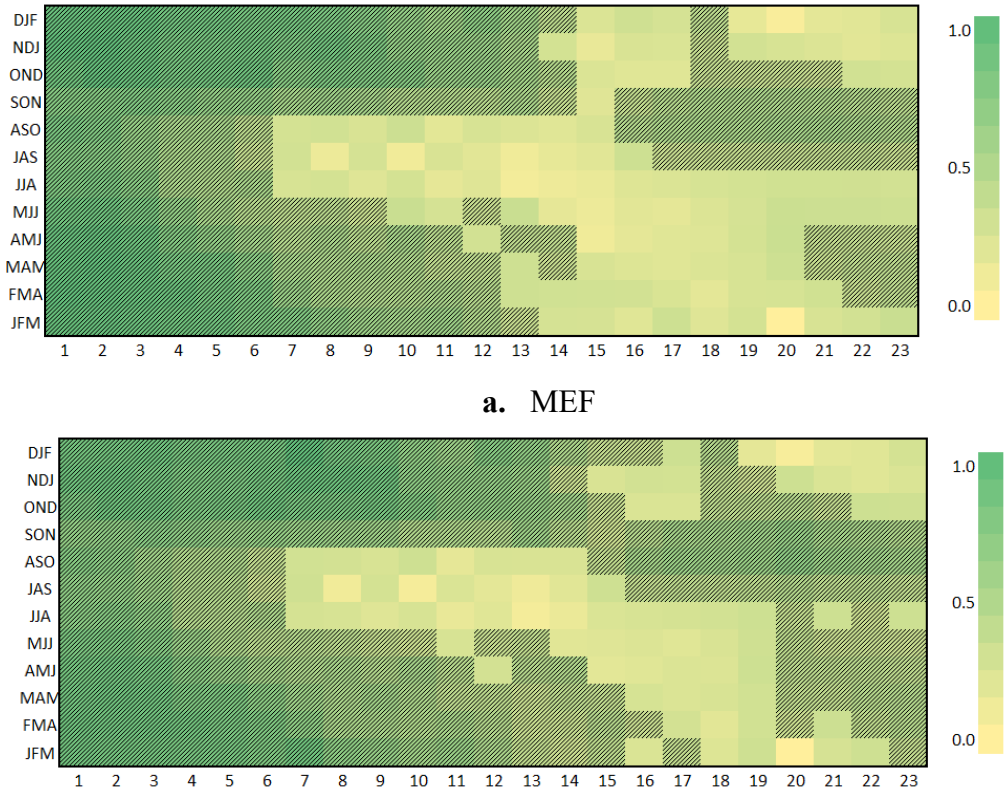
3.2. Generalizability to Other Ensemble Forecasting Models

Although the current work was devoted to the performance improvement of the MEF model through GNN-based ensemble selection, the suggested method is by no means restricted to the said architecture. In fact, the method has general applicability to any prediction system that produces a number of ensemble members as outputs, a characteristic common to state-of-the-art ENSO prediction systems, including those based on dynamical simulations, hybrid

statistical-dynamical approaches, and other deep learning architectures. In typical ensemble forecasting procedures, the final prediction is typically formed by averaging all of the ensemble members under the assumption of equal contribution from all. This assumption may not be true in reality, especially if some members have a substantially better skill in portraying the climate dynamics underlying the event. By introducing a graph-based analysis that quantifies ensemble member similarity and structural coherence, our method allows for the detection and utilization of the most reliable subsets of predictions. This general framework enables model developers to:

- Exclude anomalous or underperforming ensemble members.
- Enhance robustness and interpretability of ensemble predictions.
- Improve the skill of predictions without forcing architectural modifications to the base model.

In this way, our method offers a model-agnostic and flexible solution for ensemble refinement. It can be easily integrated into existing post-processing chains of climate prediction systems, thereby facilitating more informed decision-making in operational seasonal-to-subseasonal (S2S) forecasting applications.



a. MEF

b. EMEF

Figure 6. Correlation skill of the EMEF model. ENSO correlation for each of 276 target months (23×12). Hatched grids correlate above 0.5. The period of validation is 2001-2017.

A detailed seasonal comparison further elucidates the EMEF's superiority over the MEF across all targeted seasons, as shown in Figure . The EMEF consistently outperforms the

MEF. Both models encounter challenges associated with spring predictability barriers; however, the EMEF maintains a correlation exceeding 0.5 for nearly all targeted seasons, whereas the MEF's performance falls below this threshold for most lead months beyond 14 months. This enhancement significantly bolsters the model's predictive skill beyond 15 months, owing to the GNN design.

4. Conclusion

Long-lead forecasting of the ENSO remains a critical yet challenging goal in climate science, with profound implications for sectors such as tourism, water resource management, and disaster preparedness. While traditional numerical models offer detailed physical representations, they often demand significant computational resources. Meanwhile, artificial intelligence-based approaches have emerged as promising alternatives, enabling more efficient and accurate predictions. Building upon previous efforts like the MEF model, this study introduces a novel graph-based framework that optimizes ensemble forecasting by leveraging similarity-based clustering of individual ensemble members. By identifying and averaging a structurally coherent subset of high-performing forecasts, the EMEF method enhances predictive skill through effective noise reduction and improved ensemble coherence. This approach not only surpasses the MEF model in forecasting accuracy across nearly all seasons but also provides new insights into ensemble behavior. Its model-agnostic nature suggests broad applicability to diverse forecasting systems, including statistical, physical, and hybrid models, marking a significant step forward in the pursuit of reliable, long-term ENSO predictions.

5. References

Ali, M., Richter, S., & Ertürk, A. (2025). Graph neural networks learn emergent tissue properties from spatial molecular profiles. *Nature Communications, 16*, 8419. https://doi.org/10.1038/s41467-025-63758-8

Andrej, K. (2016). *Generative models*. OpenAI.

Blöschl, G., Hall, J., Viglione, A., et al. (2019). The changing climate both increases and decreases European river floods. *Nature, 573*, 108–111.

Cai, W., Borlace, S., & Lengaigne, M. (2014). Increasing frequency of extreme El Niño events due to greenhouse warming. *Nature Climate Change, 4*, 111–116. https://doi.org/10.1038/nclimate2100

Cai, W., McPhaden, M. J., & Grimm, A. M. (2020). Climate impacts of the El Niño–Southern Oscillation on South America. *Nature Reviews Earth & Environment, 1*, 215–231. https://doi.org/10.1038/s43017-020-0040-3

Castangia, M., Grajales, L., Aliberti, A., & Rossi, C. (2023). Transformer neural networks for interpretable flood forecasting. *Environmental Modelling & Software, 160*, 105581.

Erdmann, M., Glombitza, J., & Quast, T. (2019). Precise simulation of electromagnetic calorimeter showers using a Wasserstein generative adversarial network. *Computing and Software for Big Science, 3*(4). https://arxiv.org/abs/1807.01954

Ganji, S., Labibzadeh, A. R., Hassani, A., & Naisipour, M. (2025). Leveraging GNN to enhance the MEF method in predicting ENSO. *arXiv preprint* arXiv:2508.07410. https://doi.org/10.48550/arXiv.2508.07410

Ganji, S., Naisipour, M., Hassani, A., & Adib, A. (2025). Distillation of CNN ensemble results for enhanced long-term prediction of the ENSO phenomenon. *arXiv preprint* arXiv:2509.06227. https://doi.org/10.48550/arXiv.2509.06227

Gao, C., & Zhang, R. H. (2017). The roles of atmospheric wind and entrained water temperature in the secondary cooling of the 2010–12 La Niña event. *Climate Dynamics, 48*, 597–617.

Goodfellow, I. J., et al. (2004). Elasto-plastic fracturing model for wellbore stability using non-penetrating fluids. *ARXIV preprint* arXiv:1406.2661. https://doi.org/10.48550/arXiv.1406.2661

Ham, Y., Kim, J., & Luo, J. J. (2019). Deep learning for multi-year ENSO forecasts. *Nature, 573*, 568–572.

Hassani, A., & Javadi, M. (2025a). When AI replaces developers: Graph neural networks reveal paths from computer to management occupations. *Preprints, 2025*, 2025120212. https://doi.org/10.20944/preprints2025120212.v1

Hassani, A., & Javadi, M. (2025b). Leveraging artificial intelligence to study and forecast its impacts on employment management and wages in China's key cities. *Spectrum of Engineering and Management Sciences, 1*, 1–12. https://doi.org/10.31181/sems4120266oh

Hassani, A., Javadi, M., & Naisipour, M. (2025). The time series Informer model for stock market prediction. Preprints, 2025101842. https://doi.org/10.20944/preprints202510.1842.v1

Ho, J., & Ermon, S. (2016). Generative adversarial imitation learning. *Advances in Neural Information Processing Systems, 29*, 4565–4573. https://arxiv.org/abs/1606.03476

Isola, P., Zhu, J., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial nets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1125–1134.

Labibzadeh, M., et al. (2015). Efficiency test of the discrete least-squares meshless method in solving heat conduction problems using error estimation. *Sharif Civil Engineering, 31*(2), 31–40. https://sid.ir/paper/127886/en

L'Heureux, M. L., Levine, A. F., Newman, M., Ganter, C., Luo, J. J., Tippett, M. K., et al. (2020). ENSO prediction. In *El Niño–Southern Oscillation in a Changing Climate* (pp. 227–246). John Wiley & Sons. https://doi.org/10.1002/9781119548164.ch10

Luc, P., et al. (2016). Semantic segmentation using adversarial networks. *Advances in Neural Information Processing Systems, 29*. https://arxiv.org/abs/1611.08408

Luo, J. J., Masson, S., Behera, S. K., & Yamagata, T. (2008). Extended ENSO predictions using a fully coupled ocean–atmosphere model. *Journal of Climate, 21*, 84–93.

McPhaden, M. J., Zebiak, S. E., & Glantz, M. H. (2006). ENSO as an integrating concept in Earth science. *Science, 314*, 1740–1745.

Moazezi Khah, Tehran, A., Hassani, A., Mohajer, S., Darvishan, S., Shafiesabet, A., & Tashakkori, A. (2025). The impact of financial literacy on financial behavior and financial resilience with the mediating role of financial self-efficacy. *International Journal of Industrial Engineering and Operational Research, 7*(2), 38–55.

Mohamed, S., & Lakshminarayanan, B. (2016). Learning implicit generative models. *arXiv preprint* arXiv:1610.03483.

Mustafa, M., et al. (2019). CosmoGAN: Creating high-fidelity weak lensing convergence maps using generative adversarial networks. *Computational Astrophysics and Cosmology, 6*(1), 1. https://arxiv.org/abs/1706.02390

Naisipour, M., Afshar, M. H., Hassani, B., & Zeinali, M. (2009). An error indicator for two-dimensional elasticity problems in the discrete least-squares meshless method. *Proceedings of the 8th International Congress on Civil Engineering, Shiraz, Iran*. https://civilica.com/doc/68292

Naisipour, M., Saeedpanah, I., & Adib, A. (2025a). Multimodal deep learning for two-year ENSO forecast. *Water Resources Management, 39*, 3745–3775. https://doi.org/10.1007/s11269-025-04128-3

Naisipour, M., Saeedpanah, I., & Adib, A. (2025b). Metrics matter: A deep assessment of deep learning CNN method for ENSO forecast. *Atmospheric Research, 108545*. https://doi.org/10.1016/j.atmosres.2025.108545

Naisipour, M., Saeedpanah, I., & Adib, A. (2024). Novel deep learning method for forecasting ENSO. *Journal of Hydraulic Structures, 14*, 25–42. https://doi.org/10.22055/jhs.2024.47965.1322

Nicolas, C., Francisco, M., Gabriel, S., & Nicolas, U. A. (2020). End-to-end object detection with transformers. In *Proceedings of the European Conference on Computer Vision* (pp. 213–229).

Paganini, M., de Oliveira, L., & Nachman, B. (2017). Learning particle physics by example: Location-aware generative adversarial networks for physics synthesis. *Computing and Software for Big Science, 1*(4). https://doi.org/10.1007/s41781-017-0004-6

Reiser, P., Neubert, M., & Eberhard, A. (2022). Graph neural networks for materials science and chemistry. *Nature Communications Materials, 3*, 93. https://doi.org/10.1038/s43246-022-00315-6

Rives, A., Meier, J., Sercu, T., & Goyal, S. T. (2016). Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences, 118*(15), e2016239118.

Salamani, D., et al. (2018). Deep generative models for fast shower simulation in ATLAS. In *IEEE 14th International Conference on e-Science* (pp. 348–348).

Salimans, T., et al. (2016). Improved techniques for training GANs. *arXiv preprint* arXiv:1606.03498.

Scaife, A. A., et al. (2010). Impact of ENSO on European climate. In the *ECMWF Seminar on Predictability in the European and Atlantic Regions*, ECMWF, Shinfield Park, Reading.

Schawinski, K., et al. (2017). Generative adversarial networks recover features in astrophysical images of galaxies beyond the deconvolution limit. *Monthly Notices of the Royal Astronomical Society: Letters, 467*(1), L110–L114.

Schwaller, P., Laino, T., Gaudin, T., Bolgar, P., Hunter, C., Bekas, C., & Lee, A. (2019). Molecular transformer: A model for uncertainty-calibrated chemical reaction prediction. *ACS Central Science, 5*(9), 1572–1583. https://doi.org/10.1021/acscentsci.9b00576

Taccari, M. L., Wang, H., & Nuttall, J. E. (2024). Spatial–temporal graph neural networks for groundwater data. *Scientific Reports, 14*, 24564. https://doi.org/10.1038/s41598-024-75385-2

Talaat, F. M., & Gamel, S. A. (2022). RL-based hyper-parameter optimization algorithm (ROA) for a convolutional neural network. *Journal of Ambient Intelligence and Humanized Computing, 13*(10), 4637–4651.

Vaswani, A., Shazeer, N., & Parmar, N. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems* (pp. 5998–6008).

Wang, G. G., Cheng, H., Zhang, Y., & Yu, H. (2023). ENSO analysis and prediction using deep learning: A review. *Neurocomputing, 520*, 216–229.

Wu, S. (2025). Application of an artificial intelligence graph convolutional network in classroom grade evaluation. *Scientific Reports, 15*, 32044. https://doi.org/10.1038/s41598-025-17903-4

Ye, F., Hu, J., Huang, T., You, L., Weng, B., & Gao, J. (2022). Transformer for El Niño–Southern Oscillation prediction. *IEEE Geoscience and Remote Sensing Letters, 19*, 1–5.

Zhou, H., Zhang, S., & et al. (2021). Informer: Beyond efficient transformer for long-sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 35, pp. 11106–11115).