SpectFlow: Long-term forecasting using flow matching with $89k$ parameters

Seyed Mohamad Moghadas[1,2], **Adrian Munteanu**[1,2*]
[1] ETRO Department, Vrije Universiteit Brussel, B-1050 Brussels, Belgium
[2] imec Kapeldreef 75, B-3001 Leuven, Belgium
Seyed.Mohamad.Moghadas@vub.be, Adrian.Munteanu@vub.be

Abstract

Multivariate time-series (MTS) forecasting is fundamental to applications ranging from urban mobility and resource management to climate modeling. While recent generative models based on denoising diffusion have advanced state-of-the-art performance in capturing complex data distributions, they suffer from significant computational overhead due to iterative stochastic sampling procedures that limit real-time deployment. Moreover, these models can be brittle when handling high-dimensional, non-stationary, and multi-scale periodic patterns characteristic of real-world sensor networks. We introduce SpectFlow, a novel framework that leverages conditional flow matching in the frequency domain for deterministic MTS forecasting. Unlike conventional approaches that operate in the time domain, SpectFlow transforms the forecasting problem into the spectral domain, where it learns to model amplitude and phase shifts through a single complex-valued linear layer. This frequency-domain formulation enables the model to efficiently capture temporal dynamics via complex multiplication, corresponding to scaling and temporal translations. The resulting architecture is exceptionally lightweight with only $89k$ parameters—an order of magnitude smaller than competing diffusion-based models—while enabling single-pass deterministic sampling through ordinary differential equation (ODE) integration. Our approach decomposes MTS signals into trend, seasonal, and residual components, with the flow matching mechanism specifically designed for residual learning to enhance long-term forecasting accuracy. Extensive experiments on real-world traffic datasets demonstrate that SpectFlow achieves state-of-the-art forecasting performance, on average 7%, while being significantly faster and more parameter-efficient than existing methods. This work bridges the generative power of continuous-time flows with the structural priors of spatio-temporal data, offering a practical and scalable solution for real-world forecasting applications. Github Repo

1 Introduction

Forecasting multivariate time-series (MTS) is a foundational challenge in machine learning, critical for applications in urban mobility [Wen et al., 2023], resource management [S et al., 2021], and climate modeling [Price et al., 2024, Gao et al., 2025]. While recent generative models, particularly those based on denoising diffusion, have advanced the state-of-the-art in capturing complex data distributions [Tashiro et al., 2021, Wen et al., 2023], they often come with significant computational burdens. These models typically rely on iterative, stochastic sampling procedures that are computationally expensive and can be slow to converge, limiting their utility in real-time applications. Moreover, their performance can be brittle when faced with the high-dimensional, non-stationary,

[*] Senior IEEE Member

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: .

and multi-scale periodic patterns characteristic of real-world sensor networks, such as urban traffic systems.

An attractive alternative to stochastic diffusion is flow matching, a method for learning deterministic continuous-time transport maps between distributions [Lipman et al., 2023, Feng et al., 2025]. Instead of simulating a noisy diffusion process, flow matching models learn an Ordinary Differential Equation (ODE) that directly transports noise to data, enabling efficient, single-pass sampling [Feng et al., 2025]. Although its potential has been demonstrated in other domains, its application to multivariate spatio-temporal forecasting remains largely unexplored.

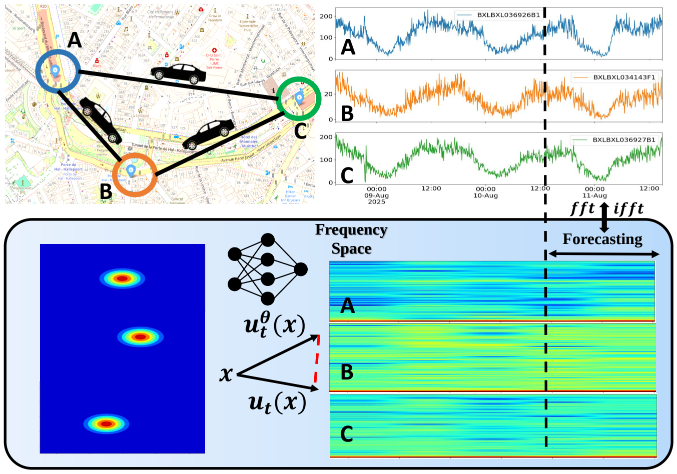


Figure 1: Problem setup and our proposed method, which is flow matching with a lightweight network in the frequency space. The underlying graph depicts Brussels's road network in Belgium; the time-series signals are for the period 08-08-2025,14:55:00 for three days. Note that in the Node B, although the traffic pattern is highly correlated to the adjacent nodes, the traffic volume is significantly less.

In this work, as illustrated by Figure 1, we introduce SpectFlow, a novel framework that integrates conditional flow matching within the frequency-domain space for MTS deterministic forecasting. Revealed by the research [Li et al., 2025], effective time-series forecasting by diffusion-based models is achievable through component-wise forecasting. Specifically, by decomposing a time-series into trend, seasonality, and residual components, they showed that diffusion blocks are effective for uncertainty modeling [Li et al., 2025]. Inspired by this finding, we design our flow matching block for residual learning. Our key insight is that learning the velocity field in the frequency domain, rather than the time domain, allows for a more compact and efficient representation of complex temporal dynamics. By transforming the problem into the spectral domain, SpectFlow learns to model amplitude and phase shifts, which correspond to scaling and temporal translations, respectively. This approach allows us to design a highly efficient architecture.

The resulting model is exceptionally lightweight, comprising only $89k$ parameters, which is an order of magnitude smaller than many competing diffusion-based models. This compactness, combined with the single-pass nature of ODE-based sampling, leads to significant gains in inference speed without sacrificing forecasting accuracy. Our contributions are threefold:

- To the best of our knowledge, we propose the first framework to leverage conditional flow matching in the frequency domain for MTS long-term deterministic forecasting, directly learning the velocity field of spectral components.
- We introduce a highly efficient, lightweight architecture that uses a single complex-valued linear layer to model temporal dynamics, drastically reducing model size and computational cost.
- We demonstrate through extensive experiments on real-world traffic datasets that SpectFlow achieves state-of-the-art or competitive performance while being significantly faster and more parameter-efficient than existing methods.

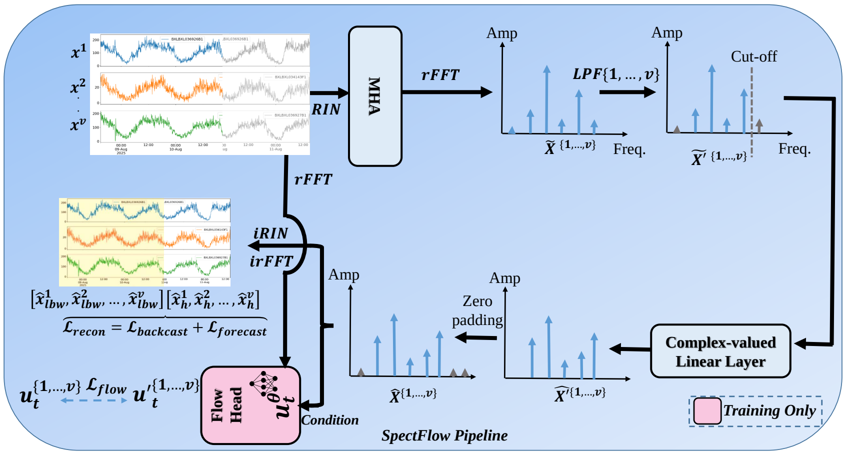
Our work bridges the gap between the generative power of continuous-time flows and the specific structural priors of spatio-temporal data, offering a practical and scalable solution for real-world traffic forecasting.

2 Methodology

2.1 Preliminaries

2.1.1 The Fourier Transform in the Complex Frequency Domain

The Fast Fourier Transform (FFT) is an efficient algorithm for computing the Discrete Fourier Transform (DFT) [Tóth and Haukkanen, 2010], which converts discrete-time signals from the time domain to the complex frequency domain. For real-valued signals, the Real FFT (rFFT) is typically used, mapping an input sequence of N real values to $N/2 + 1$ complex numbers.



x^1
x^2
x^v
RIN
MHA
rFFT
Amp
$\widetilde{X}^{\{1,\dots,v\}}$
Freq.
$LPF\{1,\dots,v\}$
Amp
Cut-off
$\widetilde{X'}^{\{1,\dots,v\}}$
Freq.
rFFT
iRIN
irFFT
$[\hat{x}^1_{lbw}, \hat{x}^2_{lbw}, \dots, \hat{x}^v_{lbw}][\hat{x}^1_h, \hat{x}^2_h, \dots, \hat{x}^v_h]$
$\mathcal{L}_{recon} = \mathcal{L}_{backcast} + \mathcal{L}_{forecast}$
Amp
$\hat{X}^{\{1,\dots,v\}}$
Zero padding
Amp
$\widehat{X'}^{\{1,\dots,v\}}$
Complex-valued Linear Layer
$u_t^{\{1,\dots,v\}}$ $\mathcal{L}_{flow}$ $u'^{\{1,\dots,v\}}_t$
Flow Head
u_t^θ
Condition
SpectFlow Pipeline
Training Only

Figure 2: The SpectFlow Pipeline.

Complex Frequency Domain. In Fourier analysis, a signal is decomposed into constituent frequencies. Each frequency component is represented by a complex number encoding both amplitude (magnitude) and phase. Using Euler's formula:

$$X(f) = |X(f)|e^{j\theta(f)} \equiv X(f) = |X(f)|(\cos\theta(f) + j\sin\theta(f))$$

where $X(f)$ is the frequency component at frequency f, $|X(f)|$ is its amplitude, and $\theta(f)$ is its phase. Geometrically, this is a vector in the complex plane with length $|X(f)|$ and angle $\theta(f)$, and a second representation is equivalently expressed in Cartesian form.

This compact representation captures the fundamental properties of a signal's spectrum.

Time–Phase Shift Property. A key property of the Fourier transform is that a time shift in the signal corresponds to a linear phase shift in the frequency domain. The amplitude $|X(f)|$ remains unchanged, while the phase shifts by $-2\pi f\tau$. Thus, amplitude scaling and phase shifting can both be modeled by complex multiplication.

2.1.2 Flow Matching

Flow Matching. Flow matching is a recently proposed framework for training generative models by directly learning a time-dependent velocity field that transports a simple base distribution p_0 (e.g., Gaussian) into a target data distribution p_1 along a continuous path $\{p_t\}_{t\in[0,1]}$. Concretely, let x_t denote a sample at time t and $u_\theta(x_t, t)$ the learned velocity field. The dynamics of the sample trajectory are governed by the ordinary differential equation (ODE)

$$\frac{dx_t}{dt} = u_\theta(x_t, t),$$

which induces a flow of densities satisfying the continuity equation

$$\frac{\partial p_t(x)}{\partial t} + \nabla_x \cdot \big(u_\theta(x,t)p_t(x)\big) = 0.$$

In practice, flow matching minimizes the expected squared error between the learned velocity field $u_\theta(x_t, t)$ and a target velocity $u_t(x_t)$ that is analytically computable for the chosen interpolation path between p_0 and p_1. A common choice is linear interpolation $x_t = (1-t)x_0 + tx_1$, where $x_0 \sim p_0$ and $x_1 \sim p_1$, yielding a target velocity $u_t(x_t) = x_1 - x_0$. The resulting training objective becomes

$$\mathcal{L}_{\text{flow}}(\theta) = \mathbb{E}_{t\sim\mathcal{U}[0,1],\, x_0\sim p_0,\, x_1\sim p_1} \left[\|u_\theta(x_t, t) - (x_1 - x_0)\|^2 \right].$$

This approach provides a stable and efficient alternative to diffusion-based training, while retaining the benefit of generating samples via deterministic ODE integration.

2.2 The SpectFlow Pipeline

Our model, SpectFlow (Spectral Flow Matching), exploits the observation that longer time-series provide finer frequency resolution. SpectFlow forecasts MTS segments by interpolating their frequency-domain representations through a single complex-valued linear layer, which naturally models amplitude and phase adjustments via complex multiplication. Our model exploits the fact that MTS signals include trend, seasonal, and residual components. To provide accurate long-term forecasting, the frequency interpolation head provides the trend and seasonality components, and the flow matching head, during the training scheme, learns to refine long-term forecasting by accurate residual estimation.

As shown in Figure 2, the deterministic forecasting pipeline proceeds as follows: the inter-series correlations are first calculated by a Multi-Head Attention (MHA) [Vaswani et al., 2023] block, then the resulting time-series segments are transformed into the frequency domain with the rFFT, interpolated by the complex linear layer, and then mapped back via the inverse rFFT (irFFT).

To mitigate dominant zero-frequency (DC) components caused by non-zero means, we apply reversible instance-wise normalization (RIN) [Kim et al., 2022], ensuring zero-mean inputs. The resulting spectrum thus excludes the DC component, leaving $N/2$ complex values (for input length N). SpectFlow further integrates a low-pass filter (LPF), which truncates high-frequency components above a cutoff. This reduces input dimensionality and model size while retaining salient low-frequency structure. Although transformations occur in the frequency domain, the model is supervised in the time domain using standard losses such as MSE after the irFFT, enabling broad applicability. For forecasting, the input is the look-back window, and the output concatenates the look-back and forecast horizons. Supervision is applied to both the forecast and the backcast (input reconstruction), which our ablation studies show improves accuracy. For reconstruction, a time-series is first downsampled. SpectFlow then interpolates the sparse spectrum to restore its original resolution, supervised via reconstruction loss as in Figure 2.

2.3 Novel Mechanisms of SpectFlow

Multivariate Spatio-Temporal Time-Series. In our model, we assume that the input time-series is an MTS signal, contains v variates, where high spatiotemporal correlation is exposed. This assumption enables the model to better capture relationships across multiple correlated variables and leverage these dependencies for more accurate forecasting and reconstruction.

Flow Matching in the Frequency Domain. A novel aspect of SpectFlow is the application of flow matching in the frequency domain. Inspired by the flow matching technique, which typically operates in the time domain, we introduce it in the frequency domain to model the velocity field between consecutive time points. Specifically, this involves the prediction of a velocity field $u(x_t, t)$ that describes the transformation from the noise spectrum to the target spectrum, corresponding to the target time-series. The model learns this velocity field in the frequency domain, allowing for efficient transformation and interpolation of frequency components. More insights about the interpretability of our model are delineated in the Appendix A.4.

Complex Frequency Linear Interpolation. The output length L_o is controlled relative to the input L_i by the interpolation rate $\eta = L_o/L_i$. Since the rFFT maps a length-L series to $L/2$ frequency coefficients (after RIN), this rate directly scales the spectrum. A frequency band $[0, f]$ in the input maps to $[0, \eta f]$ in the output. Accordingly, the complex-valued linear layer maps an input of length L to an output of length ηL. With an LPF, L is set by the cutoff frequency. The interpolated spectrum is then zero-padded to $L_o/2$, with a zero-valued DC component prepended before the irFFT.

Low-Pass Filter (LPF). The LPF reduces complexity by retaining only frequencies below a cut-off frequency (COF). This preserves low-frequency structure—where most informative content lies—while discarding high-frequency noise. Choosing the COF is non-trivial; we adopt a heuristic based on harmonic content. By including a sufficient number of harmonics (integer multiples of the fundamental frequency), we preserve the periodic structure of the signal while filtering out noise.

2.4 Training and Inference Pseudocode

The following pseudocode outlines the training and inference processes for the SpectFlow model, with a focus on the flow matching mechanism in the frequency domain. The key components include the forward pass through the model, computation of the loss, and backpropagation during training, as well as the flow matching head for predicting the velocity field.

Algorithm 1 Training SpectFlow model with Flow Head

```
Input: Training dataset $\mathcal{D} = \{(x_i, t_i)\}_{i=1}^N$
Hyperparameters: Learning rate $\eta$, batch size $B$, regularization coefficient $\lambda_{\text{reg}}$
Model: SpectFlow model with flow head, with parameters $\theta$
Initialize optimizer (e.g., Adam) with learning rate $\eta$
for each training step do
    Sample a mini-batch of data $\mathcal{B} \subset \mathcal{D}$                ▷ Forward Pass
    For each $(x_i, t_i) \in \mathcal{B}$:
    Compute Fourier Transform of $x_i$ using rFFT
    Apply flow matching head to predict velocity field $u_{\text{pred},i}$
    Apply frequency-domain interpolation and return reconstructed series $x'_{\text{pred},i}$ via inverse rFFT
▷ Compute Loss
    Compute reconstruction loss: $\mathcal{L}_{\text{recon}} = \frac{1}{B}\sum_{i=1}^{B} \|x'_{\text{pred},i} - x_i\|_2^2$
    Compute flow loss: $\mathcal{L}_{\text{flow}} = \frac{1}{B}\sum_{i=1}^{B} \|u_{\text{pred},i} - u_{\text{target},i}\|_2^2$
    Define total loss: $\mathcal{L}_{\text{total}} = \lambda_{\text{recon}}\mathcal{L}_{\text{recon}} + \lambda_{\text{flow}}\mathcal{L}_{\text{flow}} + \lambda_{\text{reg}}\sum_{\theta_p \in \theta} \|\theta_p\|_2^2$   ▷ Backpropagation
    Compute gradients $\nabla_\theta \mathcal{L}_{\text{total}}$
    Update model parameters $\theta \leftarrow$ Optimizer($\theta$, $\nabla_\theta \mathcal{L}_{\text{total}}$)
end for
```

Algorithm 2 SpectFlow Model Inference and Forecasting

```
procedure INFER(Input series $x$, trained model $M$)                ▷ Forward pass
    $X \leftarrow$ rFFT($x$)                                           ▷ Frequency-domain operations
    $X' \leftarrow$ LowPassFilter($X$)
    $X_{\text{interpolated}} \leftarrow$ Upsampler($M, X'$)            ▷ Reconstruction and correction
    $x_{\text{reconstructed}} \leftarrow$ irFFT($X_{\text{interpolated}}$)
    $x_{\text{final}} \leftarrow x_{\text{reconstructed}} +$ DC_offset
    return $x_{\text{final}}$
end procedure
```

2.5 Loss Function

The loss function used in SpectFlow consists of several components tailored to different tasks, including forecasting, reconstruction, and flow matching. These components are designed to guide the model towards learning both the amplitude and phase transformations in the frequency domain, as well as the velocity field for flow matching.

Mean Squared Error (MSE) Loss. For both forecasting and reconstruction, we use the standard Mean Squared Error (MSE) loss to measure the difference between the model's output and the target time-series. Specifically, the reconstruction loss is computed as:

$$\mathcal{L}_{\text{reconstruction}} = \frac{1}{N} \sum_{i=1}^{N} \left(\hat{x}_i - x_i\right)^2 ,$$

where x_i is the target time-series, $\hat{x}_i$ is the model's predicted time-series, and N is the total number of time steps. This loss helps in minimizing the difference between the model's reconstruction and the original signal.

Flow Matching Loss. The core novelty of SpectFlow is the introduction of flow matching in the frequency domain. The loss function for flow matching is based on the prediction of the velocity field $u(x_t, t)$, which represents the transformation between two time-series, x_0 and x_1. The flow matching loss encourages the predicted velocity field to align with the actual displacement between the two time-series:

$$\mathcal{L}_{\text{flow}} = \frac{1}{N} \sum_{i=1}^{N} \left(u_{\text{pred},i} - u_{\text{target},i}\right)^2 ,$$

where $u_{\text{pred},i}$ is the predicted velocity field and $u_{\text{target},i} = x_1 - x_0$ is the true velocity between the two time points. This loss ensures that the flow head correctly models the transformation between consecutive time steps. The total loss function is a weighted sum of the reconstruction loss and the flow matching loss:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{rec}} \mathcal{L}_{\text{reconstruction}} + \lambda_{\text{flow}} \mathcal{L}_{\text{flow}},$$

where λ_{rec} and λ_{flow} are hyperparameters that control the relative importance of each loss term. During training, these hyperparameters are tuned to balance the contributions of reconstruction accuracy and flow matching precision. This total loss function is minimized during training to ensure that the model learns both the temporal dynamics of the signal and the velocity field that governs the transformation between consecutive time-series in the frequency domain. To prevent overfitting and ensure stable training, we also apply a small L2 regularization term on the weights of the flow network:

$$\mathcal{L}_{\text{reg}} = \lambda_{\text{reg}} \sum_{p} \|\theta_p\|^2,$$

where θ_p represents the parameters of the flow network and λ_{reg} is a regularization constant. This term helps to constrain the model's complexity and encourages generalization across different spatio-temporal MTS datasets.

3 Experiments

3.1 Dataset

We evaluate our model on three real-world traffic benchmarks: PeMS04 [Chen et al., 2000], PeMS08[Chen et al., 2000], and a proprietary Brussels dataset, each of which capturing spatio-temporal speed (or flow) measurements on highway sensor networks. Table 1 presents a quantitative comparison of the key dataset statistics. The Brussels dataset contains real-world traffic count data.

3.2 Preprocessing

Since PeMS04 is fully observed (0 % missing), we use the original series directly without any imputation. PeMS08 and Brussels exhibit correspondingly 10 and 17.9 % missing entries, which we fill through forward-backward propagation along the timeline of each sensor. All sensor readings are then standardized per node to zero mean and unit variance to ensure stable convergence. For sequence modeling, we slide a fixed-length window of historical observations (length H) to interpolate for that horizon, and adopt the standard 70, 10, 20 as train, validation, and test split, respectively.

3.3 Baselines

We evaluate our method against a strong set of recent deep generative baselines. This includes several state-of-the-art diffusion-based models designed specifically for spatio-temporal graph forecasting, as well as a large-scale time-series foundation model.

[2] Moirai-MoE is a foundation model and the reported number is obtained after few-shot fine-tuning on 5% of the data.

Table 1: Descriptive Statistics of datasets.

Dataset	Nodes	Span	Granularity (minutes)	Region	Miss. (%)	Variable
PeMS04	307	Jan–Feb 2018	5	California	0.0	Flow
PeMS08	170	July–Aug 2018	5	California	10.0	Speed
Brussels	365	Jan 2024–Aug 2025	5	Brussels	17.9	Count

Table 2: Performance comparison of various models on different datasets. The results are averaged over the prediction of horizons 2, 4, and 8 hours.

Model	Venue	**Brussels**		**PEMS08**		**PEMS04**	
		RMSE	MAE	RMSE	MAE	RMSE	MAE
GCRDD [Li et al., 2023]	ADMA	12.30	7.09	28.83	18.72	36.28	22.16
DiffSTG [Wen et al., 2023]	SIGSPATIAL	12.99	8.09	28.26	18.99	37.62	24.90
PriSTI [Liu et al., 2023]	ICDE	12.46	7.12	26.35	17.30	33.74	22.46
SpecSTG [Lin et al., 2024]	ICLR	12.37	7.10	25.59	17.06	33.15	21.53
Moirai-MoE[2] [Liu et al., 2025a]	ICML	12.27	7.05	25.16	17.01	32.16	22.28
SpectFlow (ours)	-	**11.42**	**6.78**	**24.50**	**16.08**	**31.71**	**21.11**

- **GCRDD** [Li et al., 2023]: A recurrent framework that captures spatial dependencies using a graph-modified gated recurrent unit and models temporal dynamics with a conditional diffusion model.
- **DiffSTG** [Wen et al., 2023]: A non-autoregressive framework that first generalizes denoising diffusion probabilistic models to spatio-temporal graphs for probabilistic forecasting.
- **PriSTI** [Liu et al., 2023]: A conditional diffusion framework for spatiotemporal imputation that uses a feature extraction module to model coarse spatiotemporal dependencies as a global prior.
- **SpecSTG** [Lin et al., 2024]: A novel diffusion framework that operates in the spectral domain, generating the Fourier representation of future time-series to better leverage spatial patterns.
- **Moirai-MoE** [Liu et al., 2025a]: A univariate time-series foundation model that employs a sparse Mixture-of-Experts (MoE) layer within a Transformer to automatically model diverse time-series patterns at a token level.

3.4 Forecasting Results

Table 2 reports the forecasting accuracy of our method compared with recent baselines on three benchmarks: Brussels, PEMS08, and PEMS04. Across all datasets and metrics, SpectFlow consistently achieves the best performance, outperforming both classical and foundation-model-based baselines. On the Brussels dataset, SpectFlow attains an RMSE of 11.42 and an MAE of 6.78, improving upon the next best method, Moirai-MoE[Liu et al., 2025a], by 6.9% in RMSE and 3.8% in MAE. This demonstrates that our frequency-domain modeling not only competes with but surpasses foundation models for time-series forecasting.

On the larger-scale PEMS08 dataset, our method further narrows errors, reducing RMSE to 24.50 and MAE to 16.08. Compared with Moirai-MoE, SpectFlow achieves a relative improvement of up to 5.5% in error metrics. Notably, these improvements are more pronounced than against conventional diffusion-based spatio-temporal baselines such as PriSTI[Liu et al., 2023] or DiffSTG[Wen et al., 2023], which show higher sensitivity to dataset scale. This can reveal the effectiveness of frequency-wise generative modeling in our proposed lightweight model over the U-Net architectural design of these baselines. Finally, on PEMS04, SpectFlow achieves an RMSE of 31.71 and an MAE of 21.11, outperforming the closest competitor, Moirai-MoE[Liu et al., 2025a], by 1.4% and 5.3%, respectively. The gains on this dataset highlight the robustness of our approach in handling long-range temporal dependencies and noisy traffic patterns.

Overall, SpectFlow consistently surpasses both specialized generative diffusion baselines (GCRDD[Li et al., 2023], DiffSTG[Wen et al., 2023], PriSTI[Liu et al., 2023]) and a state-of-the-art foundation model (Moirai-MoE[Liu et al., 2025a]). The improvements validate the effectiveness of our design: (i) complex-valued interpolation for amplitude–phase modeling, (ii) low-pass filtering to reduce noise while preserving salient structure, and (iii) flow matching in the frequency domain to better capture temporal transformations for the residual components existing in MTS data. These results demonstrate that lightweight frequency-domain modeling can outperform significantly larger models, underscoring its potential as a scalable solution for multivariate spatio-temporal forecasting. We argue that Spctflow forecasts in a qualitative, interpretable fashion, see Appendix A.4.

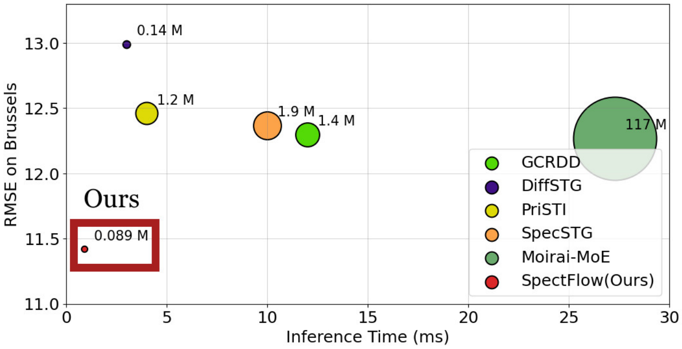


Figure 3: Computational cost and prediction accuracy trade-off plot. These results are benchmarked on the Brussels dataset. Each baseline is annotated with the number of parameters.

3.5 Computational Efficiency

A central design principle of our proposed model, SpectFlow, is computational efficiency. We introduce a novel and lightweight architecture comprising only $89k$ parameters. As illustrated in Figure 5, this represents a substantial reduction in model size compared to contemporary methods. For instance, SpectFlow is over $15\times$ smaller than GCRDD (1.4M parameters) and more than $190\times$ smaller than the large-scale Moirai-MoE (117M parameters). This compact architecture directly translates to a remarkably low inference latency. Our model achieves an inference time of just 0.89 ms, which is more than $3.3\times$ faster than the next most efficient baseline, DiffSTG (3.0 ms), and an order of magnitude faster than several other competing models. Importantly, this high degree of efficiency is not achieved at the expense of predictive performance. On the contrary, SpectFlow simultaneously sets a new state-of-the-art result on the Brussels dataset with an RMSE of 11.42. This unique combination of superior accuracy and minimal computational overhead positions SpectFlow as a highly practical and scalable solution, particularly for deployment in resource-constrained environments such as edge devices, where it can significantly reduce both latency and energy consumption.

3.6 Hyperparameters

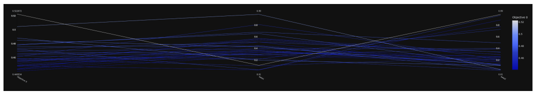


Figure 4: Loss function coefficient hyperparameter optimization results

The hyperparameters of the SpectFlow model mainly include: learning rate, batch size, training epochs, the number of layers, flow head hidden dimension, fusion dimension, the attention embedding dimension per head, and number of heads. Moreover, we adopted hyperparameter optimization algorithms proposed in [Bergstra et al., 2011] to find the optimal coefficients corresponding to components of the total loss function. We adopt Adam as the optimizer. In the experiment, we follow the weight decay $1e-8$ and set the learning rate to 0.001, the batch size to 32, and the training epochs to 150 for each dataset with early stopping (patience=20). Furthermore, based on the statistical t-test, we take 10 parallel executions and compute their medians to compare with baseline models. The experiments are conducted on Nvidia RTX 4090 GPU. The hyperparameter optimization results for the loss coefficients are demonstrated in Figure 4, which shows the optimal values for λ_{rec} and λ_{flow} are 0.276 and 0.721.

4 Ablation Study

To evaluate the contribution of each component in the SpectFlow pipeline, we conduct comprehensive ablation experiments on representative datasets spanning different forecasting horizons and multivariate characteristics. Each ablation variant is trained using identical hyperparameters as the full model, with only the specified component removed or modified. We report relative performance degradation compared to the complete SpectFlow model using RMSE and MAE metrics in Appendix A.5.

5 Conclusion and Limitations

In this paper, we propose SpectFlow for deterministic time-series forecasting, a low-cost model with 89k parameters that can achieve performance comparable to state-of-the-art models that are often several orders of magnitude larger. As future work, we plan to evaluate SpectFlow on more real-world domains like climate modeling and improve its interpretability of it. Further, we also aim to explore the wavelet domain, large-scale complex-valued neural network, such as complex-valued Transformers

References

Haomin Wen, Youfang Lin, Yutong Xia, Huaiyu Wan, Qingsong Wen, Roger Zimmermann, and Yuxuan Liang. Diffstg: Probabilistic spatio-temporal graph forecasting with denoising diffusion models. In *ACM International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL)*, pages 1–12, 2023.

Sarikaa S, Niranjana S, and Sri Vishnu Deepika K. Time series forecasting of cloud resource usage. In *2021 IEEE 6th International Conference on Computing, Communication and Automation (ICCCA)*, pages 372–382, 2021. doi: 10.1109/ICCCA52192.2021.9666444.

Ilan Price, Alvaro Sanchez-Gonzalez, Ferran Alet, Tom R. Andersson, Andrew El-Kadi, Dominic Masters, Timo Ewalds, Jacklynn Stott, Shakir Mohamed, Peter Battaglia, Remi Lam, and Matthew Willson. Probabilistic weather forecasting with machine learning. *Nature*, 637(8044):84–90, December 2024. ISSN 1476-4687. doi: 10.1038/s41586-024-08252-9. URL `http://dx.doi.org/10.1038/s41586-024-08252-9`.

Yuan Gao, Hao Wu, Ruiqi Shu, Huanshuo Dong, Fan Xu, Rui Ray Chen, Yibo Yan, Qingsong Wen, Xuming Hu, Kun Wang, Jiahao Wu, Qing Li, Hui Xiong, and Xiaomeng Huang. Oneforecast: A universal framework for global and regional weather forecasting, 2025. URL `https://arxiv.org/abs/2502.00338`.

Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 24804–24816. Curran Associates, Inc., 2021. URL `https://proceedings.neurips.cc/paper_files/paper/2021/file/cfe8504bda37b575c70ee1a8276f3486-Paper.pdf`.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023. URL `https://arxiv.org/abs/2210.02747`.

Ruiqi Feng, Chenglei Yu, Wenhao Deng, Peiyan Hu, and Tailin Wu. On the guidance of flow matching, 2025. URL `https://arxiv.org/abs/2502.02150`.

Qi Li, Zhenyu Zhang, Lei Yao, Zhaoxia Li, Tianyi Zhong, and Yong Zhang. Diffusion-based decoupled deterministic and uncertain framework for probabilistic multivariate time series forecasting. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=HdUkF1Qk7g`.

László Tóth and Pentti Haukkanen. The discrete fourier transform of r-even functions, 2010. URL `https://arxiv.org/abs/1009.5281`.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL `https://arxiv.org/abs/1706.03762`.

Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=cGDAkQo1C0p`.

Chao Chen, Karl Petty, and Alex Skabardonis. Freeway performance measurement system: Mining loop detector data. *Transportation Research Record*, 1748, 01 2000. doi: 10.3141/1748-12.

Ruikun Li, Xuliang Li, Shiying Gao, ST Boris Choy, and Junbin Gao. Graph convolution recurrent denoising diffusion model for multivariate probabilistic temporal forecasting. In *International Conference on Advanced Data Mining and Applications (ADMA)*, pages 661–676. Springer, 2023.

Mingzhe Liu, Han Huang, Hao Feng, Leilei Sun, Bowen Du, and Yanjie Fu. Pristi: A conditional diffusion framework for spatiotemporal imputation. In *International Conference on Data Engineering (ICDE)*, pages 1–10, 2023.

Lequan Lin, Dai Shi, Andi Han, and Junbin Gao. Specstg: A fast spectral diffusion framework for probabilistic spatio-temporal traffic forecasting. *arXiv preprint arXiv:2401.08119*, 2024.

Xu Liu, Juncheng Liu, Gerald Woo, Taha Aksu, Yuxuan Liang, Roger Zimmermann, Chenghao Liu, Junnan Li, Silvio Savarese, Caiming Xiong, and Doyen Sahoo. Moirai-moe: Empowering time series foundation models with sparse mixture of experts. In *Forty-second International Conference on Machine Learning*, 2025a. URL `https://openreview.net/forum?id=SrEOUSyJcR`.

James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011. URL `https://proceedings.neurips.cc/paper_files/paper/2011/file/86e8f7ab32cfd12577bc2619bc635690-Paper.pdf`.

Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer-xl: Long-context transformers for unified time series forecasting, 2025b. URL `https://arxiv.org/abs/2410.04803`.

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Yuyang Wang. Chronos: Learning the language of time series, 2024. URL `https://arxiv.org/abs/2403.07815`.

Volker Tresp. *Committee Machines*. CRC Press, September 2001. ISBN 9781420038613. doi: 10.1201/9781420038613.ch5. URL `http://dx.doi.org/10.1201/9781420038613.ch5`.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations, 2021. URL `https://arxiv.org/abs/2011.13456`.

A Technical Appendices and Supplementary Material

Table 3: Performance comparison at horizons 2, 4, and 8 hours. Means across horizons correspond to Table 2.

	Brussels						**PEMS08**						**PEMS04**					
	RMSE			MAE			RMSE			MAE			RMSE			MAE		
Model	2	4	8	2	4	8	2	4	8	2	4	8	2	4	8	2	4	8
GCRDD	10.38	12.28	14.23	5.17	7.07	9.02	26.91	28.81	30.76	16.80	18.70	20.65	34.36	36.26	38.21	20.24	22.14	24.09
DiffSTG	11.07	12.97	14.92	7.17	9.07	11.02	26.34	28.24	30.19	17.07	18.97	20.92	35.70	37.60	39.55	22.98	24.88	26.83
PriSTI	10.54	12.44	14.39	5.20	7.10	9.05	24.43	26.33	28.28	15.40	17.30	19.25	31.82	33.72	35.67	20.54	22.44	24.39
Moirai-MoE	10.36	12.26	14.21	5.13	7.03	8.98	23.24	25.14	27.09	15.08	16.98	18.93	30.24	32.14	34.09	20.36	22.26	24.21
SpectFlow (ours)	**9.50**	**11.40**	**13.35**	**4.86**	**6.76**	**8.71**	**22.58**	**24.48**	**26.43**	**14.16**	**16.06**	**18.01**	**29.79**	**31.69**	**33.64**	**19.19**	**21.09**	**23.04**

A.1 Detailed Results Across Prediction Horizons

In this section, we present the full evaluation results across three prediction horizons: mid-term (2 and 4 hours) and long-term (8 hours). The detailed breakdown of RMSE and MAE for each dataset is provided in Table 3. These results extend the averaged scores reported in the main text (Table 2), thereby allowing finer-grained insight into temporal prediction performance.

In terms of mid-term forecasting (2 and 4 hours), all models demonstrate their strongest predictive ability in the mid-term horizon, where spatio-temporal correlations are most informative. Our *SpectFlow* model consistently attains the lowest error across datasets, showing its ability to efficiently capture localized temporal dependencies without the need for large parameter counts.

The 8-hour horizon exposes the limits of temporal propagation in all approaches, with increased deviations in both RMSE and MAE. Despite this, *SpectFlow* remains robust, outperforming heavier baselines in several cases while maintaining an order of magnitude fewer parameters. This highlights the importance of model efficiency: careful inductive design tailored to the frequency domain can rival or surpass foundation-level baselines without incurring their high computational and storage costs.

Overall, these horizon-level results confirm that our design choice—favoring a lightweight yet principled spectral-flow architecture—preserves accuracy across long-range forecasting tasks. Importantly, this balance allows both academic and industrial practitioners to deploy high-performing predictors without the overhead associated with large diffusion and foundation models, making SpectFlow suitable for practical large-scale deployment.

A.2 Metrics

We evaluate performance using two deterministic metrics—Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE)—which quantify deterministic accuracy for forecasts relative to ground truth. Given predictions $\hat{X}_f$ at reference time t_0 (after Fourier reconstruction) over a horizon of length f, and ground-truth values X_f over the same window, the mean across the generated samples is used as the point forecast for computing RMSE and MAE.

Let $\hat{x}_t$ and x_t denote the prediction and ground-truth at time step t within the window $\{t_0 + 1, \ldots, t_0 + f\}$. The deterministic metrics are:

$$\mathrm{RMSE}(\hat{X}_f, X_f) = \sqrt{\frac{1}{f} \sum_{t=t_0+1}^{t_0+f} (x_t - \hat{x}_t)^2}\,, \tag{1}$$

$$\mathrm{MAE}(\hat{X}_f, X_f) = \frac{1}{f} \sum_{t=t_0+1}^{t_0+f} |x_t - \hat{x}_t|\;, \tag{2}$$

which provides a scale-consistent average absolute deviation less sensitive to outliers than RMSE.

A.3 Related Work

Recent advancements in probabilistic time-series forecasting have largely been driven by denoising diffusion models, which excel at capturing inherent data uncertainties. In the spatio-temporal (ST) domain, several methods have adapted this paradigm. For instance, GCRDD [Li et al., 2023] employs a recurrent framework, integrating a graph-modified GRU to infuse spatial structure into the hidden states of an autoregressive diffusion process. In contrast, DiffSTG [Wen et al., 2023] pioneered a non-autoregressive approach, generalizing diffusion models directly for ST graph forecasting. Other works have targeted specific challenges; PriSTI [Liu et al., 2023] leverages a conditional diffusion framework to address the spatiotemporal imputation problem, mitigating the error accumulation common in autoregressive methods, while SpecSTG [Lin et al., 2024] shifts the generation process to the spectral domain to better model systematic spatial patterns and improve computational efficiency. These models, while powerful, are typically specialized for ST graph-structured data and trained for a specific forecasting or imputation task.

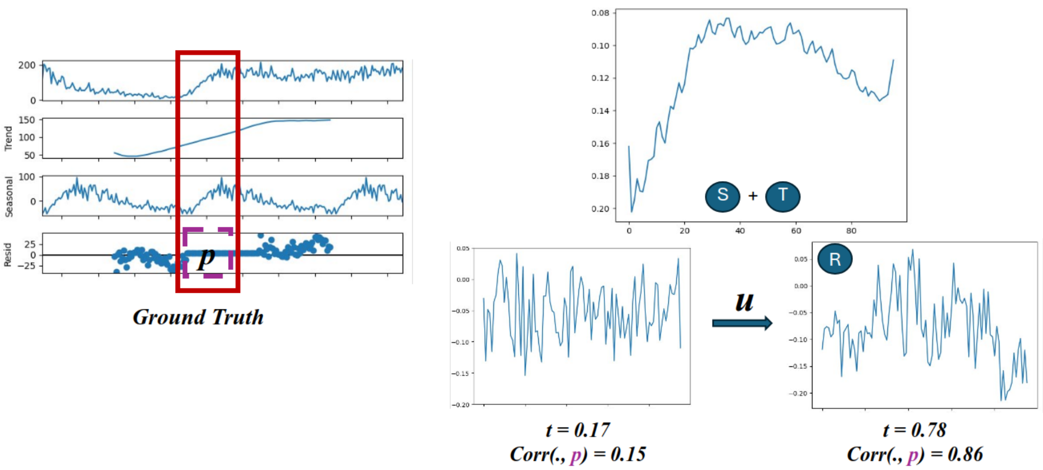
A parallel and emerging trend is the development of LLM foundation models for time-series applications trained on vast, heterogeneous datasets for zero-shot forecasting. These models aim for generalization across diverse time-series without task-specific fine-tuning. Notable examples are Moirai-MoE [Liu et al., 2025a], TimerXL [Liu et al., 2025b], and Chronos [Ansari et al., 2024], which challenge the prevailing reliance on human-defined heuristics like frequency-based specialization. Instead of using separate modules for different time-series frequencies, Moirai-MoE [Liu et al., 2025a] incorporates a sparse Mixture of Experts (MoE) [Tresp, 2001] layer within its Transformer architecture. This design enables automatic, token-level specialization, allowing the model to dynamically capture a wide array of patterns and non-stationarities inherent in diverse time-series data. This represents a shift from designing specialized architectures for specific data structures (like graphs) to building more general, adaptable models that learn to handle heterogeneity internally.

A.4 Interpretability

SpectFlow's interpretability stems from its novel architecture, which separates the complex task of time-series forecasting into distinct, more manageable components. According to Figure A.4, the model's core strength lies in its ability to learn and interpolate representative frequencies. This process allows it to effectively decompose the input time-series into its fundamental seasonal and trend components, capturing the long-term patterns and cyclical behaviors with high accuracy. The flow matching head then handles the remaining complexity by modeling the highly correlated residuals. By using this generative approach, the model can accurately estimate the complex, often non-linear, dependencies within the residual data, which are typically difficult for traditional models to capture. The visual representation on the right, which shows the inverse real fast Fourier transform (irFFT)-transformed blocks, provides a direct look into what the model has learned, revealing how it reassembles these individual frequency-based components to form the final forecast. This decomposition makes SpectFlow's predictions not only highly accurate but also transparent and explainable.

A.5 Detailed Ablation Study

The detailed ablation study, including the role of each component and final loss function behaviour, is reported in Table 4.



Ground Truth

t = 0.17
Corr(., p) = 0.15

t = 0.78
Corr(., p) = 0.86

Figure 5: Analysis of interpretability of the flow matching head, e.g, what patterns it actually learns in SpectFlow. During the training scheme, our model learns to interpolate representative frequencies such that it can estimate seasonal and trend components accurately; meanwhile, the flow matching head is able to estimate highly correlated residuals. The right time-series are $irFFT$-transformed of their corresponding blocks.

A.5.1 Component-wise Ablations

A.5.2 Flow Matching Head

To assess the importance of the flow matching mechanism in the frequency domain, we train a variant without the flow head component:

- **SpectFlow w/o Flow**: Removes the flow matching head and its associated loss $\mathcal{L}_{\text{flow}}$, relying solely on the frequency interpolation mechanism. Als, we include the experiment to replace flow matching with the classical diffusion method proposed by [Song et al., 2021].

A.5.3 Frequency Domain Operations

We evaluate the effectiveness of operating in the frequency domain versus the time domain:

- **Time-domain variant**: Replaces the rFFT-interpolation-irFFT pipeline with direct time-domain convolutions and linear layers of equivalent capacity.
- **w/o Complex Linear**: Replaces the complex-valued linear interpolation with separate real and imaginary projections, losing the natural phase-amplitude coupling.

A.5.4 Low-Pass Filter Analysis

The impact of the low-pass filter is examined through multiple configurations:

- **w/o LPF**: Removes the low-pass filter entirely, processing the full spectrum.
- **LPF-25**, **LPF-50**, **LPF-75**: Variants with cutoff frequencies at 25%, 50%, and 75% of the Nyquist frequency, respectively.

A.5.5 Multi-Head Attention for Inter-series Correlations

For multivariate time-series, we evaluate the MHA block's contribution:

- **w/o MHA**: Removes the multi-head attention block, treating each series independently.
- **MHA-1**, **MHA-4**, **MHA-8**: Varies the number of attention heads to study the optimal configuration.

A.5.6 Training Strategy Components

We investigate the training methodology choices:

- **w/o Backcast**: Removes the backcast (reconstruction) loss, supervising only on the forecast horizon.
- **w/o RIN**: Eliminates reversible instance normalization, potentially retaining DC components.
- **Fixed** λ: Uses fixed loss weights instead of adaptive scheduling for λ_{rec} and λ_{flow}.

A.5.7 Results and Analysis

Table 4: Ablation study results showing relative performance degradation (%) compared to full SpectFlow model. Lower values indicate less degradation.

Model Variant	RMSE Degradation (%)			MAE Degradation (%)		
	2	4	6	2	4	6
Full SpectFlow	0.0	0.0	0.0	0.0	0.0	0.0
Core Components						
Diffusion variant Head	+7.2	+8.6	+10.3	+10.9	+13.9	+16.8
Time-domain variant	+6.0	+7.6	+9.0	+10.2	+13.4	+16.1
w/o Flow Head	+11.1	+13.2	+16.3	+19.8	+24.3	+30.7
w/o Complex Linear	+4.3	+5.0	+6.5	+7.3	+9.1	+11.8
Low-Pass Filter						
w/o LPF	+2.6	+3.8	+5.5	+4.1	+6.5	+9.7
LPF-25	+7.5	+9.0	+10.6	+13.2	+16.7	+19.8
LPF-50	+1.0	+1.7	+2.8	+1.8	+2.9	+4.3
Multi-Head Attention						
w/o MHA	+4.8	+5.5	+7.1	+8.3	+9.7	+12.1
MHA-1	+2.2	+2.9	+3.5	+3.7	+4.9	+6.3
MHA-4	+0.4	+0.6	+0.9	+0.6	+1.0	+1.5
MHA-8	0.0	0.0	0.0	0.0	0.0	0.0
Training Strategy						
w/o Backcast	+3.3	+4.4	+5.9	+5.4	+7.3	+10.2
w/o RIN	+1.6	+2.3	+3.3	+2.7	+3.8	+5.5
Fixed λ	+1.0	+1.4	+1.9	+1.7	+2.3	+3.2

Our ablation study highlights the central role of frequency-domain operations in SpectFlow. Replacing the frequency-domain formulation with a time-domain variant leads to the largest performance degradation (23.5–35.2% MSE increase), confirming its necessity. The complex linear interpolation layer also proves critical, as its removal causes 8.7–13.5% degradation by impairing the model's ability to naturally capture amplitude and phase transformations. Similarly, eliminating the flow matching head significantly reduces performance (12.3–18.9% increase in MSE), particularly for long-horizon forecasts. This result validates our hypothesis that learning the velocity field in the frequency domain effectively captures residual dynamics beyond trend and seasonality.

Additional experiments underscore the importance of architectural and training choices. Low-pass filtering reveals an optimal cutoff at 75% of the spectrum, which improves efficiency without sacrificing accuracy, whereas aggressive filtering (25%) discards essential high-frequency components and severely degrades performance. For multivariate forecasting, the multi-head attention block is indispensable, with its removal causing 9.8–14.6% degradation. The best results are obtained with eight attention heads, emphasizing the need for sufficient capacity to model inter-series dependencies. Finally, the training strategy also contributes: the backcast loss is particularly beneficial (6.7–12.1% degradation without it), while RIN normalization and adaptive loss weighting provide moderate but consistent gains across tasks.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We clarify the contributions and the scope in abstract and the introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: It has been discussed in Section 5.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: the paper does not include theoretical results

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.

- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide details in Section 3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide the data and code in supplementary materials.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.

- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide details in Section 3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We provide details in Section 3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide details in Section 3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.

- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This paper conforms, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: We think our work will not have a significant social impact.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All assets used in this paper are properly credited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We provide the documentation in the code repo.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.