AdaPTS: Adapting Univariate Foundation Models to Probabilistic Multivariate Time Series Forecasting

Abdelhakim Benechehab[*12], **Vasilii Feofanov**[1], **Giuseppe Paolo**[1], **Albert Thomas**[1], **Maurizio Filippone**[3], **Balázs Kégl**[1]

[1] Huawei Noah's Ark Lab, Paris, France

[2] Department of Data Science, EURECOM

[3] Statistics Program, KAUST

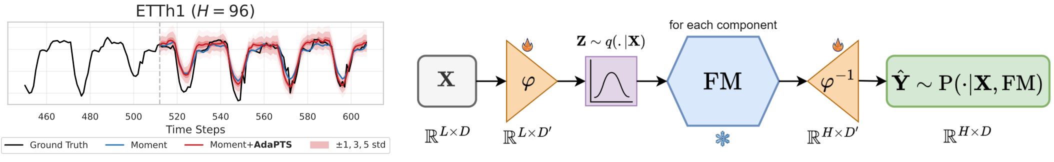
* Correspondence to abdelhakim.benechehab@gmail.com

Abstract

Pre-trained foundation models (FMs) have shown exceptional performance in univariate time series forecasting tasks. However, several practical challenges persist, including managing intricate dependencies among features and quantifying uncertainty in predictions. This study aims to tackle these critical limitations by introducing **adapters**—feature-space transformations that facilitate the effective use of pre-trained univariate time series FMs for multivariate tasks. Adapters operate by projecting multivariate inputs into a suitable latent space and applying the FM independently to each dimension in a zero-shot manner. Inspired by the literature on representation learning and partially stochastic Bayesian neural networks, we present a range of adapters and optimization/inference strategies. Experiments conducted on both synthetic and real-world datasets confirm the efficacy of adapters, demonstrating substantial enhancements in forecasting accuracy and uncertainty quantification compared to baseline methods. Our framework, **AdaPTS**, positions adapters as a modular, scalable, and effective solution for leveraging time series FMs in multivariate contexts, thereby promoting their wider adoption in real-world applications. We release the code at https://github.com/abenechehab/AdaPTS.

1 Introduction

Time series forecasting is a well-established machine learning problem that involves analyzing sequential data to predict future trends based on historical patterns. Two key challenges frequently arise in this context: (a) time series are often multivariate, incorporating multiple descriptive features (Wei, 2019), and (b) estimating the uncertainty of a forecast is equally important, requiring probabilistic model outputs (Gneiting & Katzfuss, 2014). These challenges are particularly relevant in real-world



(a) (b)

Figure 1: (a) Augmenting Moment time series foundation model with the **AdaPTS** framework provides *probabilistic* and *more accurate* predictions. (b) **The AdaPTS framework:** The input time series is transformed through a feature space transformation φ that maps into a stochastic latent space. The prediction is then conducted using a pre-trained FM before transforming back the predicted, now distribution, to the original feature space. The fire symbol indicate trainable weights while the snowflake implicates that the parameters of the FM are kept frozen.

applications where risk assessment depends on reliable forecasts, such as healthcare (Jones & Spiegelhalter, 2012), finance (Groen et al., 2013), energy management (Zhang et al., 2014; Nowotarski & Weron, 2018), and weather prediction (Palmer, 2012; Bi et al., 2023).

Existing foundation models (FMs) for time series forecasting, such as `Chronos` (Ansari et al., 2024), are typically trained for univariate forecasting tasks due to tractability constraints, as the wide range of real world time series problems typically have different numbers of features. Even without discretization, handling multivariate time series directly within these models (`Moment` (Goswami et al., 2024), `Moirai` (Liu et al., 2024)) remains computationally challenging due to the high-dimensional dependencies among features. This limitation raises a fundamental question: how can we leverage existing pre-trained univariate FMs to enable probabilistic forecasting for multivariate time series?

To address this, we introduce **`AdaPTS`**, a novel framework designed to augment FMs with probabilistic adapters. As illustrated in Figure 1, **`AdaPTS`** applies a stochastic feature transformation that maps the input time series into a latent space, where predictions are made using the frozen FM. Our framework sets itself apart from existing literature by enforcing an invertibility constraint on the adapter, allowing predictions to be transformed back into the original feature space. Beyond enhancing forecasting accuracy, the integration of stochasticity into the adapter's latent representation ensures that the model captures uncertainty, thereby improving both calibration and robustness.

Our approach leads to several novel insights and contributions, which we summarize as follows:

1. **Multivariate FM adaptation.** We introduce a principled methodology for adapting existing pre-trained univariate FMs to multivariate probabilistic forecasting, resulting in the **`AdaPTS`** framework.
2. **Theoretical foundations of adapters.** We provide a theoretical analysis to support the necessity of adapters, starting with the analytically tractable case of linear adapters and linear FMs. We then build on the literature on partially stochastic Bayesian neural networks to introduce probabilistic adapters.
3. **Empirical validation.** We conduct extensive experiments on multivariate time series forecasting benchmarks, demonstrating that our approach improves forecasting accuracy baseline methods. We also analyze the interpretability of the learned latent representation and show that adapters enable cost-effective adaptation by reducing the dimensionality of the feature space.

2 AdaPTS: Adapters for Probabilistic Multivariate Time Series Forecasting

2.1 Problem Setup

Consider a multivariate long-term time series forecasting task, represented by: a data matrix $\mathbf{X} \in \mathbb{R}^{L \times D}$ where L is the context window size and D is the multivariate time series dimensionality, and a target matrix $\mathbf{Y} \in \mathbb{R}^{H \times D}$, where H is the forecasting horizon. We denote by $\mathbf{x}_d \in \mathbb{R}^{L \times 1}$ (respectively $\mathbf{y}_d \in \mathbb{R}^{H \times 1}$) the d-th component of the input (respectively target) multivariate time series.

Our goal is to use a frozen pre-trained univariate time series foundation model denoted as $f_{\mathrm{FM}} : \mathbb{R}^{L \times 1} \to \mathbb{R}^{H \times 1}$ (Fig. 1b) and exploit the information stored in its weights to achieve the best forecasting performance, measured by the mean squared error (MSE) loss:

$$\mathcal{L} = \|\mathbf{Y} - f_{\mathrm{FM}}(\mathbf{X})\|_{\mathrm{F}}^2 \tag{1}$$

On multivariate time series, for simplicity, we denote by $f_{\mathrm{FM}}(\mathbf{X})$ the application of f_{FM} to each channel independently, in which case the loss can be written as: $\frac{1}{D}\sum_{d=1}^{D} \|\mathbf{y}_d - f_{\mathrm{FM}}(\mathbf{x}_d)\|_2^2$.

We now formally define an adapter, a tool by means of which we aim to best use the foundation model f_{FM} for multivariate forecasting:

Definition 2.1 (adapter)**.** An adapter is a feature-space transformation $\varphi : \mathbb{R}^D \rightarrow \mathbb{R}^{D'}$ that is applied to the data prior to the foundation model[1]. The forecast is then obtained by transforming the predictions back to the original feature space:

$$\hat{\mathbf{Y}}(\mathbf{X}; \varphi) = \varphi^{-1}\big(f_{\mathrm{FM}}(\varphi(\mathbf{X}))\big)$$

According to this definition, an adapter is valid only if the inverse transformation $\varphi^{-1} : \mathbb{R}^{D'} \rightarrow \mathbb{R}^D$, such that $\forall \mathbf{x} \in \mathbb{R}^D$, $\varphi^{-1} \circ \varphi(\mathbf{x}) = \mathbf{x}$, is well-defined on $\mathbb{R}^{D'}$. In the rest of the paper, we relax this condition by naming the direct transformation as *encoder* ($\varphi \triangleq \mathrm{enc}$), and respectively, the inverse transformation as *decoder* ($\varphi^{-1} \triangleq \mathrm{dec}$). In this case, the predictions obtained after the application of the adapter become: $\hat{\mathbf{Y}}(\mathbf{X}; \mathrm{enc}, \mathrm{dec}) = \mathrm{dec}\big(f_{\mathrm{FM}}(\mathrm{enc}(\mathbf{X}))\big)$.

2.2 Families of adapters

Linear AutoEncoders. In addition to the setup introduced in Eq. (13), we extend Linear AutoEncoders to provide a simple yet effective method for dimensionality reduction while preserving the temporal relationships within time series data. In this more general case, the encoder compresses the multivariate time series $\mathbf{X}$ into a potentially lower-dimensional representation $\mathbf{Z} = \mathbf{X}\mathbf{W}_{\theta_{\mathrm{enc}}}$, where $W \in \mathbb{R}^{D \times D'}$ is the linear transformation matrix, and $D' \leq D$. The decoder reconstructs the forecast to the original feature space after prediction as $\hat{\mathbf{Y}} = f_{\mathrm{FM}}(\mathbf{Z})\mathbf{W}_{\theta_{\mathrm{dec}}}$. Finally, the parameters of the encoder θ_{enc} and the decoder θ_{dec} are jointly optimized to minimize the objective in Eq. (2).

Deep non-linear AutoEncoders. Deep non-linear AutoEncoders extend their linear counterparts by employing multiple layers of non-linear transformations. The encoder maps the input $\mathbf{X}$ to a latent space $\mathbf{Z} = \mathrm{enc}(\mathbf{X}; \theta_{\mathrm{enc}})$, where enc is parameterized by a deep neural network. Similarly, the decoder reconstructs the predictions of the foundation model in the latent space: $\hat{\mathbf{Y}} = \mathrm{dec}(f_{\mathrm{FM}}(\mathbf{Z}); \theta_{\mathrm{dec}})$.

Besides AutoEncoders, Normalizing Flows (Kobyzev et al., 2021) such as `RealNVP` (Dinh et al., 2017) are a valid choice in the context of adapters, thanks to their inherently invertible nature. However, their training may be challenging due to various optimization related concerns. We defer a discussion on Normalizing Flows as adapters to Appendix C.

2.3 Probabilistic Adapters

Variational AutoEncoders. Following the Bayesian perspective on adapters, `VAE` assume a prior distribution over the latent representation $\mathbf{Z}$, typically $\mathcal{N}(0, \mathbf{I})$. The encoder then outputs parameters of the posterior distribution $q_\phi(\mathbf{Z}|\mathbf{X})$, and in our context, the decoder generates reconstructions of predictions $\hat{\mathbf{Y}} \sim p_\theta(\mathbf{Y}|\mathbf{X}, f_{\mathrm{FM}}(\mathbf{Z}))$ where θ parametrize a likelihood model p. We then define the training objective of the `VAE`, which brings together the forecasting loss and a regularization term, in a similar way to the *evidence lower bound* (ELBO) (Kingma & Welling, 2013) objective:

Proposition 2.2 (`VAE` adapter training objective)**.** *The training objective for the* `VAE` *adapter is the maximization of an* ELBO*-like lower bound on the marginal likelihood of the target* $\mathbf{Y}$*:*

$$\begin{aligned}\log p_\theta(\mathbf{Y}|\mathbf{X}, f_{\mathrm{FM}}) \geq & \mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X})}\left[\log p_\theta(\mathbf{Y}|\mathbf{X}, f_{\mathrm{FM}}(\mathbf{Z}))\right] \\ & - \mathrm{KL}\left(q_\phi(\mathbf{Z}|\mathbf{X}) \,\|\, p(\mathbf{Z})\right),\end{aligned}$$

where KL *denotes the Kullback-Leibler divergence.*

The derivation of this lower bound and a discussion on the implications of each term of the loss are deferred to Appendix B.4.

Remark 2.3. In practice, we use the Gaussian likelihood as our likelihood model: $p_\theta(\mathbf{Y}|\mathbf{X}, f_{\mathrm{FM}}(\mathbf{Z})) = \mathcal{N}(\mathbf{Y}; \hat{\mathbf{Y}}, \sigma^2\mathbf{I})$, with $\hat{\mathbf{Y}} = \mathrm{dec}_\theta(f_{\mathrm{FM}}(\mathbf{Z}))$. In this case the forecasting loss term boils down to the MSE objective in Eq. (2) up to a multiplicative and additive noise-related constants: $\log \mathcal{N}(\mathbf{Y}; \hat{\mathbf{Y}}, \sigma^2\mathbf{I}) = -\frac{1}{2\sigma^2}\|\mathbf{Y} - \hat{\mathbf{Y}}\|_{\mathrm{F}}^2 - \frac{HD}{2}\log(2\pi\sigma^2)$. Notice that one can also learn a model of the noise where $\mathrm{dec}_\theta(f_{\mathrm{FM}}(\mathbf{Z})) = [\mu_\theta(\mathbf{Y}|\mathbf{X}, f_{\mathrm{FM}}(\mathbf{Z})), \sigma_\theta(\mathbf{Y}|\mathbf{X}, f_{\mathrm{FM}}(\mathbf{Z}))]$.

[1] In practice, φ is applied on matrices $\mathbf{X}$ in $\mathbb{R}^{L \times D}$. This denotes the application of φ on each row of $\mathbf{X}$.

Dataset	H	No adapter	with adapter				
		Moment	PCA	LinearAE	dropoutLAE	LinearVAE	VAE
ETTh1	96	0.411$_{\pm 0.012}$	0.433$_{\pm 0.001}$	0.402$_{\pm 0.002}$	**0.395$_{\pm 0.003}$**	0.400$_{\pm 0.001}$	0.404$_{\pm 0.001}$
	192	**0.431$_{\pm 0.001}$**	0.440$_{\pm 0.000}$	0.452$_{\pm 0.002}$	0.446$_{\pm 0.001}$	0.448$_{\pm 0.002}$	**0.431$_{\pm 0.001}$**
Illness	24	2.902$_{\pm 0.023}$	2.98$_{\pm 0.001}$	2.624$_{\pm 0.035}$	2.76$_{\pm 0.061}$	2.542$_{\pm 0.036}$	**2.461$_{\pm 0.008}$**
	60	3.000$_{\pm 0.004}$	3.079$_{\pm 0.000}$	3.110$_{\pm 0.127}$	2.794$_{\pm 0.015}$	**2.752$_{\pm 0.040}$**	2.960$_{\pm 0.092}$
Weather	96	0.177$_{\pm 0.010}$	0.176$_{\pm 0.000}$	0.169$_{\pm 0.000}$	**0.156$_{\pm 0.001}$**	0.161$_{\pm 0.001}$	0.187$_{\pm 0.001}$
	192	0.202$_{\pm 0.000}$	0.208$_{\pm 0.001}$	**0.198$_{\pm 0.001}$**	0.200$_{\pm 0.001}$	0.204$_{\pm 0.000}$	0.226$_{\pm 0.000}$
ExchangeRate	96	**0.130$_{\pm 0.011}$**	0.147$_{\pm 0.000}$	0.167$_{\pm 0.013}$	**0.130$_{\pm 0.011}$**	0.243$_{\pm 0.039}$	0.455$_{\pm 0.010}$
	192	**0.210$_{\pm 0.002}$**	0.222$_{\pm 0.000}$	0.304$_{\pm 0.005}$	0.305$_{\pm 0.013}$	0.457$_{\pm 0.020}$	0.607$_{\pm 0.021}$

Table 1: Performance comparison between the baseline Moment model without adapters against different adapter architectures (PCA, LinearAE, dropoutLinearAE, LinearVAE, and VAE), for multivariate long-term forecasting with different horizons H. We display the average test MSE ± standard error obtained on 3 runs with different seeds. **Best** results are in bold, with lower values indicating better performance.

Remark 2.4. The KL divergence regularization term can be multiplied by a scaling factor β to control the disentanglement—independence of the latent representation components. This results in β-VAE (Higgins et al., 2017), which is what we use in practice while referring to it as the VAE adapter throughout the paper.

Dropout as approximate VI. Dropout (Srivastava et al., 2014) can be interpreted as a form of variational inference, where a variational distribution is imposed over the weights of a neural network (Gal & Ghahramani, 2016). Specifically, applying dropout during training corresponds to approximating a posterior over the weights using a Bernoulli distribution. This perspective allows the deterministic models introduced in Section 2.2, such as Linear AutoEncoders, to be transformed into probabilistic models by introducing stochasticity through dropout.

3 Experiments & Results

3.1 Time series forecasting

Baseline. We compare our method against the vanilla application of the foundation model Moment $_{\text{small}}$ from the Moment family of models (Goswami et al., 2024). This means that for each dataset, we apply Moment $_{\text{small}}$ independently to each feature. Additionally, we compare our learning-based adapters against PCA, an adapter that has been used in the literature for model-based reinforcement learning (Benechehab et al., 2025) and time series classification (Feofanov et al., 2024; 2025).

AdaPTS improves the performance of Moment . We present the forecasting error measured by the Mean Squared Error (MSE) in Table 1 and the Mean Absolute Error (MAE) in Appendix E. On the ETTh1 dataset with a prediction horizon $H = 96$, all adapter-based variants outperform the baseline Moment model, with dropoutLinearAE achieving the best performance, showing an 8% improvement. Similar results are observed for the Illness dataset, where all adapters improve over the baseline. Notably, the VAE achieves a significant 15% improvement, reducing the MSE from 2.902 to 2.461 at $H = 24$. In the Weather dataset, the dropoutLinearAE adapter shows the best improvement across all adapter architectures for $H = 96$, while its deterministic counterpart, LinearAE, takes the lead at $H = 192$. The results on the ExchangeRate dataset are mixed, with some adapters matching the baseline performance (dropoutLinearAE at $H = 96$) while others show degraded performance, particularly at a longer prediction horizon ($H = 192$), which is also

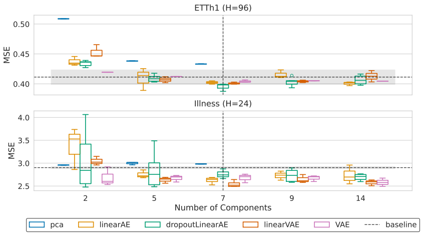


Figure 2: Impact of the number of components on model performance. The dashed line indicates Moment performance without adapters, the shaded area its standard deviation, and the vertical line the number of original features.

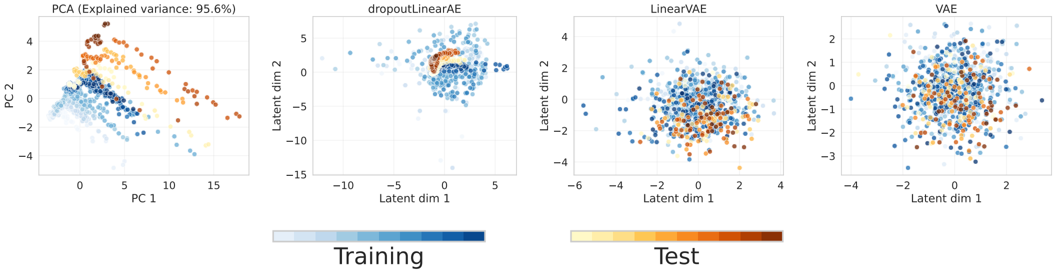


Figure 3: Visualization of the latent representation obtained by different adapters on Illness($H = 24$). Shaded colors indicate the time dimension, with lighter colors representing earlier timesteps.

observed for the ETTh1 dataset. Overall, **AdaPTS** improves the forecasting accuracy of Moment in 5 out of the 8 considered tasks, matches its performance in 2, and degrades performance in 1 task.

Dimensionality Reduction. Fig. 2 illustrates the impact of varying latent space dimensions on forecasting performance across different adapters. For the ETTh1 dataset with a 96-step horizon, all adapter architectures achieve optimal performance at 7 components (matching the original feature count), with MSE values consistently lower than the baseline. Notably, at just 5 components, all adapters (except the PCA baseline) match the baseline score, demonstrating the suitability of our framework for low-resource setups through dimensionality reduction. The Illness dataset ($H = 24$) presents more compelling results, as the VAE adapter achieves significantly optimal performance with only 2 components, underscoring the potential of our approach for cost-effective adaptation of time series foundation models.

Interpretability of the latent representations. Fig. 3 compares the representation learning capabilities of different adapters on the Illness($H = 24$) dataset, focusing on their ability to distinguish between training and test data. To visualize the raw dataset, we employ PCA for dimensionality reduction, retaining only two principal components, which is justified by the 95.6% explained variance. When representing the training and test datasets in the space of the first two principal components, we observe a clear distribution shift, potentially complicating the forecasting task for the baseline foundational model. In contrast, using **AdaPTS** results in well-overlapping Gaussian distributions for the training and test data in the latent space. This demonstrates our framework's ability to enforce a structured, isotropic representation that mitigates distribution shift. This effect is particularly pronounced with the VAE adapter and, to a lesser extent, with LinearVAE and dropoutLinearAE.

On the calibration of the probabilistic adapters. To evaluate the calibration of our adapter-based probabilistic forecasters, we use quantile calibration as depicted in the reliability diagram in Fig. 4. In an ideal scenario, a well-calibrated probabilistic forecast should align with the red dashed diagonal, indicating that the empirical proportion of observations falls within the predicted quantiles at the expected rate. The overall conclusion is that we observe a gradual deviation from ideal calibration as the prediction horizon increases (darker shades). While early prediction horizons display reasonably well-calibrated predictions, longer-horizon forecasts systematically underestimate uncertainty, as shown by the curve falling below the diagonal. This indicates that observed values exceed predicted quantiles more frequently than expected, suggesting that the predictive distribution becomes too narrow, resulting in overconfident forecasts.

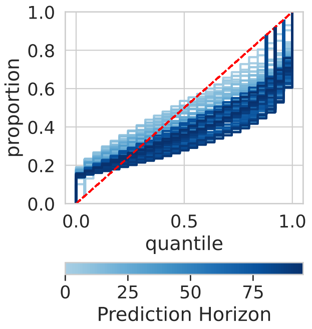


Figure 4: Reliability diagram for the first feature of the ETTh1 ($H = 96$) dataset using LinearVAE.

4 Conclusion

In this paper, we investigate how pre-trained univariate time series foundation models can be adapted for probabilistic multivariate forecasting. To address this challenge, we introduce the **AdaPTS** framework. Our method offers a novel approach to training feature space transformations that facilitate uncertainty quantification and enhance the performance of baseline foundation models. Through a series of experiments, we demonstrate that our framework improves forecasting accuracy, provides reasonably well-calibrated uncertainty estimates, reduces inference cost through dimensionality reduction, and offers interpretable feature space latent representations.

Reproducibility Statement

In order to ensure reproducibility we will release the code at https://github.com/abenechehab/AdaPTS, once the paper has been accepted. The implementation details and hyperparameters are listed in Appendix D.2.

References

Ansari, A. F., Stella, L., Turkmen, C., Zhang, X., Mercado, P., Shen, H., Shchur, O., Rangapuram, S. S., Arango, S. P., Kapoor, S., et al. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*, 2024.

Benechehab, A., Hili, Y. A. E., Odonnat, A., Zekri, O., Thomas, A., Paolo, G., Filippone, M., Redko, I., and Kégl, B. Zero-shot model-based reinforcement learning using large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=uZFXpPrwSh.

Bi, K., Xie, L., Zhang, H., Chen, X., Gu, X., and Tian, Q. Accurate medium-range global weather forecasting with 3d neural networks. *Nature*, 619(7970):533–538, 2023.

Chen, S.-A., Li, C.-L., Arik, S. O., Yoder, N. C., and Pfister, T. TSMixer: An all-MLP architecture for time series forecast-ing. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=wbpxTuXgm0.

Chen, T., Fox, E., and Guestrin, C. Stochastic Gradient Hamiltonian Monte Carlo. In Xing, E. P. and Jebara, T. (eds.), *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pp. 1683–1691, Bejing, China, 22–24 Jun 2014. PMLR. URL https://proceedings.mlr.press/v32/cheni14.html.

Cowen-Rivers, A., Lyu, W., Tutunov, R., Wang, Z., Grosnit, A., Griffiths, R.-R., Maravel, A., Hao, J., Wang, J., Peters, J., and Bou Ammar, H. Hebo: Pushing the limits of sample-efficient hyperparameter optimisation. *Journal of Artificial Intelligence Research*, 74, 07 2022.

Das, A., Kong, W., Sen, R., and Zhou, Y. A decoder-only foundation model for time-series forecasting. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=jn2iTJas6h.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using real nvp, 2017. URL https://arxiv.org/abs/1605.08803.

Feofanov, V., Ilbert, R., Tiomoko, M., Palpanas, T., and Redko, I. User-friendly foundation model adapters for multivariate time series classification. *arXiv preprint arXiv:2409.12264*, 2024.

Feofanov, V., Wen, S., Alonso, M., Ilbert, R., Guo, H., Tiomoko, M., Pan, L., Zhang, J., and Redko, I. Mantis: Lightweight calibrated foundation model for user-friendly time series classification. *arXiv preprint arXiv:2502.15637*, 2025.

Gal, Y. and Ghahramani, Z. Dropout As a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pp. 1050–1059. JMLR.org, 2016. URL http://dl.acm.org/citation.cfm?id=3045390.3045502.

Gelman, A., Carlin, J., Stern, H., Dunson, D., Vehtari, A., and Rubin, D. *Bayesian Data Analysis*. Chapman and Hall/CRC, United States, 3rd ed edition, 2013. ISBN 9781439840955.

Gneiting, T. and Katzfuss, M. Probabilistic forecasting. *Annual Review of Statistics and Its Application*, 1(1):125–151, 2014.

Goswami, M., Szafer, K., Choudhry, A., Cai, Y., Li, S., and Dubrawski, A. Moment: A family of open time-series foundation models. In *International Conference on Machine Learning*, 2024.

Graves, A. Practical variational inference for neural networks. In Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., and Weinberger, K. (eds.), *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011. URL https://proceedings.neurips.cc/paper_files/paper/2011/file/7eb3c8be3d411e8ebfab08eba5f49632-Paper.pdf.

Groen, J. J., Paap, R., and Ravazzolo, F. Real-time inflation forecasting in a changing world. *Journal of Business & Economic Statistics*, 31(1):29–44, 2013.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Sy2fzU9gl.

Ilbert, R., Odonnat, A., Feofanov, V., Virmaux, A., Paolo, G., Palpanas, T., and Redko, I. Samformer: Unlocking the potential of transformers in time series forecasting with sharpness-aware minimization and channel-wise attention. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235. PMLR, 2024. URL https://proceedings.mlr.press/v235/ilbert24a.html.

Jin, M., Wang, S., Ma, L., Chu, Z., Zhang, J. Y., Shi, X., Chen, P.-Y., Liang, Y., Li, Y.-F., Pan, S., and Wen, Q. Time-LLM: Time series forecasting by reprogramming large language models. In *International Conference on Learning Representations (ICLR)*, 2024.

Jones, H. E. and Spiegelhalter, D. J. Improved probabilistic prediction of healthcare performance indicators using bidirectional smoothing models. *Journal of the Royal Statistical Society Series A: Statistics in Society*, 175(3):729–747, 2012.

Kim, T., Kim, J., Tae, Y., Park, C., Choi, J.-H., and Choo, J. 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.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013. URL https://api.semanticscholar.org/CorpusID:216078090.

Kobyzev, I., Prince, S. J., and Brubaker, M. A. Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11): 3964–3979, November 2021. ISSN 1939-3539. doi: 10.1109/tpami.2020.2992934. URL http://dx.doi.org/10.1109/TPAMI.2020.2992934.

Lai, G., Chang, W.-C., Yang, Y., and Liu, H. Modeling long- and short-term temporal patterns with deep neural networks, 2018. URL https://arxiv.org/abs/1703.07015.

Li, S. C.-X. and Marlin, B. M. A scalable end-to-end gaussian process adapter for irregularly sampled time series classification. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/9c01802ddb981e6bcfbec0f0516b8e35-Paper.pdf.

Li, Y., Ye, J., Wen, X., Xu, G., Wang, J., and Liu, X. Padapter: Adapter combined with prompt for image and video classification. *Image and Vision Computing*, 154:105395, 2025. ISSN 0262-8856. doi: https://doi.org/10.1016/j.imavis.2024.105395. URL https://www.sciencedirect.com/science/article/pii/S0262885624005006.

Liaw, R., Liang, E., Nishihara, R., Moritz, P., Gonzalez, J. E., and Stoica, I. Tune: A research platform for distributed model selection and training, 2018. URL https://arxiv.org/abs/1807.05118.

Liu, X., Liu, J., Woo, G., Aksu, T., Liang, Y., Zimmermann, R., Liu, C., Savarese, S., Xiong, C., and Sahoo, D. Moirai-moe: Empowering time series foundation models with sparse mixture of experts. *arXiv preprint arXiv:2410.10469*, 2024.

Ma, J., Thomas, V., Hosseinzadeh, R., Kamkari, H., Labach, A., Cresswell, J. C., Golestan, K., Yu, G., Volkovs, M., and Caterini, A. L. Tabdpt: Scaling tabular foundation models, 2024. URL https://arxiv.org/abs/2410.18164.

Max Planck Institute. Weather dataset, 2021. URL https://www.bgc-jena.mpg.de/wetter/.

Nie, Y., Nguyen, N. H., Sinthong, P., and Kalagnanam, J. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Jbdc0vTOcol.

Nowotarski, J. and Weron, R. Recent advances in electricity price forecasting: A review of probabilistic forecasting. *Renewable and Sustainable Energy Reviews*, 81:1548–1568, 2018.

Palmer, T. Towards the probabilistic earth-system simulator: A vision for the future of climate and weather prediction. *Quarterly Journal of the Royal Meteorological Society*, 138(665):841–861, 2012.

Pan, J., Lin, Z., Zhu, X., Shao, J., and Li, H. St-adapter: Parameter-efficient image-to-video transfer learning. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 26462–26477. Curran Associates, Inc., 2022.

Papamarkou, T., Skoularidou, M., Palla, K., Aitchison, L., Arbel, J., Dunson, D., Filippone, M., Fortuin, V., Hennig, P., Hernández-Lobato, J. M., Hubin, A., Immer, A., Karaletsos, T., Khan, M. E., Kristiadi, A., Li, Y., Mandt, S., Nemeth, C., Osborne, M. A., Rudner, T. G. J., Rügamer, D., Teh, Y. W., Welling, M., Wilson, A. G., and Zhang, R. Position: Bayesian deep learning is needed in the age of large-scale ai. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Rasul, K., Ashok, A., Williams, A. R., Ghonia, H., Bhagwatkar, R., Khorasani, A., Bayazi, M. J. D., Adamopoulos, G., Riachi, R., Hassen, N., Biloš, M., Garg, S., Schneider, A., Chapados, N., Drouin, A., Zantedeschi, V., Nevmyvaka, Y., and Rish, I. Lag-llama: Towards foundation models for probabilistic time series forecasting, 2024.

Sharma, M., Farquhar, S., Nalisnick, E., and Rainforth, T. Do Bayesian neural networks need to be fully stochastic? In Ruiz, F., Dy, J., and van de Meent, J.-W. (eds.), *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 7694–7722. PMLR, 25–27 Apr 2023. URL https://proceedings.mlr.press/v206/sharma23a.html.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56): 1929–1958, 2014. URL http://jmlr.org/papers/v15/srivastava14a.html.

Tian, Z., Peisong, N., Xue, W., Liang, S., and Rong, J. One Fits All: Power general time series analysis by pretrained lm. In *NeurIPS*, 2023.

Tran, B.-H., Rossi, S., Milios, D., and Filippone, M. All you need is a good functional prior for Bayesian deep learning. 23(1), 2022. ISSN 1532-4435.

U.S. Centers for Disease Control and Prevention. Fluview: Flu activity & surveillance, 2024. URL https://gis.cdc.gov/grasp/fluview/fluportaldashboard.html.

Wei, W. W. *Multivariate time series analysis and applications*. John Wiley & Sons, 2019.

Wu, H., Xu, J., Wang, J., and Long, M. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=J4gRj6d5Qm.

Yang, A. X., Robeyns, M., Wang, X., and Aitchison, L. Bayesian low-rank adaptation for large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=FJiUyzOF1m.

Yin, D., Hu, L., Li, B., and Zhang, Y. Adapter is all you need for tuning visual tasks, 2023. URL https://arxiv.org/abs/2311.15010.

Zeng, A., Chen, M., Zhang, L., and Xu, Q. Are transformers effective for time series forecasting? 2023.

Zhang, W., Ye, J., Li, Z., Li, J., and Tsung, F. Dualtime: A dual-adapter multimodal language model for time series representation, 2024. URL https://arxiv.org/abs/2406.06620.

Zhang, Y. and Yan, J. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The eleventh international conference on learning representations*, 2023.

Zhang, Y., Wang, J., and Wang, X. Review on probabilistic forecasting of wind power generation. *Renewable and Sustainable Energy Reviews*, 32:255–270, 2014.

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., and Zhang, W. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *The Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Virtual Conference*, volume 35, pp. 11106–11115. AAAI Press, 2021.

Appendix

Outline. The related works are referenced in Appendix A. In Appendix B, we provide the theoretical foundations of our framework, notably the linear case analysis (B.1) and the probabilistic adapters approach (B.3). We then provide a perspective on Normalizing Flows as adapters in Appendix C. The experimental setup is presented in Appendix D, including all the implementation details in Appendix D.2. Finally we showcase some additional results and ablation studies in Appendix E.

Table of Contents

A Related work **11**

B Theoretical foundations **11**

B.1 The linear case 11

B.2 Working example 13

B.3 Probabilistic adapters 14

B.4 Proof of Proposition 2.2 14

C Normalizing Flows **15**

D Experimental setup **16**

D.1 Datasets 16

D.2 Implementation details 16

E Additional results **17**

E.1 `Moment` applied to synthetic data. 17

E.2 Mean Absolute Error 17

E.3 Ablation studies 18

A Related work

Time Series Foundational Models. Over the past two years, a plethora of foundation models have been proposed with a particular focus on time series forecasting. Some of these models like GPT4TS (Tian et al., 2023) and Time-LLM (Jin et al., 2024) ”reprogram” a Large Language Model to the forecasting setting by freezing most of its layers and fine-tuning additional time series-specific modules to a new downstream task. The majority of these time series FMs including Lag-Llama (Rasul et al., 2024), Chronos (Ansari et al., 2024), Moirai (Liu et al., 2024), TimesFM (Das et al., 2024) and Moment (Goswami et al., 2024) are trained from scratch on a large volume of time series data.

Adapters. The multivariate setting presents a significant challenge for time series FMs, as different tasks involve varying numbers of channels[2]. To the best of our knowledge, the only model that naturally accommodates any number of channels is Moirai (Liu et al., 2024), which, however, suffers from high computational demand due to processing all channels flattened in the transformer simultaneously, leading to a quadratic memory complexity w.r.t. to the number of channels. Most foundation models, instead, treat each one of these independently, which, as noted by Feofanov et al. (2024; 2025), remains computationally expensive when full fine-tuning is required. For classification tasks with hundreds or thousands of features, they demonstrated that simple adapters like the rotation matrix obtained through Principal Components Analysis (PCA) mitigate this issue. At the same time, Benechehab et al. (2025) showed that PCA preserves channel interactions by learning new disentangled components. However, in both cases, PCA provided little improvement over independent processing, leaving room for further enhancements. In the context of tabular regression, foundation models such as (Ma et al., 2024, TabDPT) also use PCA to adapt to a variable number of features.

Less related to our work, Li & Marlin (2016) use a Gaussian process adapter in the context of irregular time series classification. In other domains, adapters have been used for multimodal (text-time series) representation learning (Zhang et al., 2024) and computer vision (Li et al., 2025; Yin et al., 2023; Pan et al., 2022).

B Theoretical foundations

B.1 The linear case

In this section we consider the definition of adapters provided in Definition 2.1. We note that in the literature, there exist alternatives to adapt to the multivariate setting (Zhang & Yan, 2023; Tian et al., 2023), but we have chosen this family of adapters due to their high flexibility as: (a) any foundation model can be plugged-in, (b) no requirement of fine-tuning due to feature-level transformations (Feofanov et al., 2024; 2025), (c) adaptation to the computation budget by defining the number of encoded channels.

Optimality of an adapter. In order for an adapter to be useful, it has to achieve a lower forecasting error than the identity baseline. In fact, the loss defined in Eq. (1) corresponds to the forecasting loss obtained by using an adapter implementing the identity matrix $\mathbf{I}$. Therefore, we define the optimality of the adapter based on improving the forecasting error of the identity baseline:

$$\mathcal{L} \geq \mathcal{L}(\varphi) = \|\mathbf{Y} - \varphi^{-1}\big(f_{\mathrm{FM}}(\varphi(\mathbf{X}))\big)\|_{\mathrm{F}}^2$$

The purpose of this section is to study the optimization problem that the adapter φ is aiming to solve:

$$\varphi^* = \arg\min_{\varphi} \|\mathbf{Y} - \varphi^{-1}\big(f_{\mathrm{FM}}(\varphi(\mathbf{X}))\big)\|_{\mathrm{F}}^2 \tag{2}$$

Under mild assumptions on the adapter function class and the backbone foundation model f_{FM}, we aim at characterizing the optimal solution φ^* and prove that it realizes the optimality condition: $\mathcal{L}(\varphi^*) \leq \mathcal{L}$.

[2] Throughout the paper, the words *features*, *channels*, and *components* are used interchangeably to refer to the number of variates in a multivariate time series, represented as D in **??**.

We first consider the linear case where we constrain the adapter φ to the class of linear transformations, parametrized by a matrix $\mathbf{W}_\varphi \in \mathbb{R}^{D\times D}$: $\varphi(\mathbf{X}) = \mathbf{X}\mathbf{W}_\varphi$.

Assumption B.1. $\mathbf{W}_\varphi$ has full rank: $\text{rank}(\mathbf{W}_\varphi) = D$, insuring its invertibility.

Assumption B.2. For ease of derivation, we consider a similar linear parametrization for the foundation model: $f_{\text{FM}}(\mathbf{X}) = \mathbf{W}_{\text{FM}}^\top \mathbf{X} + \mathbf{b}_{\text{FM}}\mathbf{1}^\top$ where $\mathbf{W}_{\text{FM}} \in \mathbb{R}^{L\times H}$, $\mathbf{b}_{\text{FM}} \in \mathbb{R}^H$, and $\mathbf{1}$ a vector of ones of dimension D.

Proposition B.3 (Optimal linear adapter)**.** *Under Assumption B.1 and Assumption B.2, the* closed-form *solution of the problem:*

$$\mathcal{L}(\mathbf{W}_\varphi) = \|\mathbf{Y} - \left(\mathbf{W}_{FM}^\top \mathbf{X}\mathbf{W}_\varphi + \mathbf{b}_{FM}\mathbf{1}^\top\right)\mathbf{W}_\varphi^{-1}\|_F^2 \tag{3}$$

writes as:

$$\mathbf{W}_\varphi^* = (\mathbf{B}^\top\mathbf{A})^+\mathbf{B}^\top\mathbf{B} \tag{4}$$

where $\mathbf{W}_\varphi^* = \arg\min_{\mathbf{W}_\varphi \in \mathcal{GL}_D(\mathbb{R})} \mathcal{L}(\mathbf{W}_\varphi)$*,* $\mathbf{A} = \mathbf{Y} - \mathbf{W}_{FM}^\top\mathbf{X}$*,* $\mathbf{B} = \mathbf{b}_{FM}\mathbf{1}^\top$*, and* $(\mathbf{B}^\top\mathbf{A})^+$ *denoting the* pseudo-inverse *operator.*

Proof. We begin by expanding the loss function:

$$\begin{aligned}\mathcal{L}(\mathbf{W}_\varphi) &= \|\mathbf{Y} - (\mathbf{W}_{\text{FM}}^\top\mathbf{X}\mathbf{W}_\varphi + \mathbf{b}_{\text{FM}}\mathbf{1}^\top)\mathbf{W}_\varphi^{-1}\|_F^2 \\ &= \|\mathbf{A} - \mathbf{B}\mathbf{W}_\varphi^{-1}\|_F^2\end{aligned}$$

where $\mathbf{A} = \mathbf{Y} - \mathbf{W}_{FM}^\top\mathbf{X}$ and $\mathbf{B} = \mathbf{b}_{FM}\mathbf{1}^\top$. Expanding the Frobenius norm:

$$\mathcal{L}(\mathbf{W}_\varphi) = \text{Tr}\left((\mathbf{A} - \mathbf{B}\mathbf{W}_\varphi^{-1})^\top(\mathbf{A} - \mathbf{B}\mathbf{W}_\varphi^{-1})\right)$$

Taking the gradient with respect to $\mathbf{W}_\varphi^{-1}$ yields:

$$\frac{\partial\mathcal{L}}{\partial\mathbf{W}_\varphi^{-1}} = -2\mathbf{B}^\top\mathbf{A} + 2\mathbf{B}^\top\mathbf{B}\mathbf{W}_\varphi^{-1}$$

Knowing that $\mathbf{W}_\varphi$ is invertible, We have that: $\frac{\partial\mathcal{L}}{\partial\mathbf{W}_\varphi} = -\mathbf{W}_\varphi^{-\top}\frac{\partial\mathcal{L}}{\partial\mathbf{W}_\varphi^{-1}}\mathbf{W}_\varphi^{-\top}$

hence

$$\frac{\partial\mathcal{L}}{\partial\mathbf{W}_\varphi} = -2\mathbf{W}_\varphi^{-T}\left(\mathbf{B}^\top\mathbf{A} - \mathbf{B}^\top\mathbf{B}\mathbf{W}_\varphi^{-1}\right)\mathbf{W}_\varphi^{-T}.$$

Setting $\frac{\partial\mathcal{L}}{\partial\mathbf{W}_\varphi} = 0$ and multiplying both sides by $\mathbf{W}_\varphi^\top$, we obtain:

$$\mathbf{B}^\top\mathbf{A} = \mathbf{B}^\top\mathbf{B}\mathbf{W}_\varphi^{-1}.$$

Multiplying both sides by $\mathbf{W}_\varphi$:

$$\mathbf{B}^\top\mathbf{A}\mathbf{W}_\varphi = \mathbf{B}^\top\mathbf{B}.$$

Finally applying the pseudo-inverse to solve for $\mathbf{W}_\varphi$ gives our final result:

$$\mathbf{W}_\varphi^* = (\mathbf{B}^\top\mathbf{A})^+\mathbf{B}^\top\mathbf{B}.$$

Given the convexity of $\mathcal{L}(\mathbf{W}_\varphi)$ (which follows from the convexity of the Frobenius norm $\|\cdot\|_{\text{F}}^2$, the inverse operation, and an affine transformation), we conclude that $\mathbf{W}_\varphi^*$ is a global solution for Eq. (3).

Remark B.4. We make use of the pseudo-inverse due to the current construction of the matrix $\mathbf{B}$ (with identical rows) which implies that the product $\mathbf{B}^\top\mathbf{A}$ is degenerate. To bypass this limitation and further ensure the invertibility of $\mathbf{W}_\varphi^*$, we can revisit the definition of the foundation model in Assumption B.2 to include channel dependent biases and ensure a full rank matrix $\mathbf{B}$.

□

Remark B.5. In this case, the fact that the matrix $\mathbf{B} = \mathbf{b}_{FM}\mathbf{1}^\top$ have identical columns renders the matrix $\mathbf{B}^\top\mathbf{A}$ degenerate (with $rank(\mathbf{B}^\top\mathbf{A}) = 1$). In practice, we add a positive constant to the diagonal in order to numerically stabilize the matrix inversion: $\mathbf{W}^*_\varphi = (\mathbf{B}^\top\mathbf{A} + \lambda\mathbf{I})^{-1}\mathbf{B}^\top\mathbf{B}$, with $\lambda > 0$. In Appendix B.2 we show that we are able to reach an optimal solution regardless of this added regularization.

B.2 Working example

Synthetic data. Our synthetic dataset comprises a multivariate time series with several independent and other linearly correlated channels (Fig. 5), designed to evaluate a linear feature-space transformation. The data generation process creates five (*uncorrelated*) base signals—sinusoids with distinct frequencies, amplitudes, and *i.i.d.* noise— and derives eight additional channels through linear combinations of these bases with additive Gaussian noise of different magnitude ($\sigma \in (0.1, 0.2, 0.5)$). This construction provides a controlled environment where the ground truth relationship between channels is known: the underlying data manifold is effectively five-dimensional, but the observed eight-dimensional multivariate time series includes varying levels of noise and linear mixing.

Randomly generated linear FMs. The experimental setup in Fig. 5 consists in randomly sampling the linear parameters of a toy foundation model: $\mathbf{W}_{\text{FM}}$ and $\mathbf{b}_{\text{FM}}$. To simulate a realistic scenario, we use Glorot-uniform initialization distribution as it would be the case in neural network-based architectures. We then compute the closed-form solution $\mathbf{W}^*_\varphi$ on raw data $\mathbf{X}$, and compare the resulting loss value with the baseline (using the identity matrix $\mathbf{I}$ as adapter) and the PCA-only adapter.

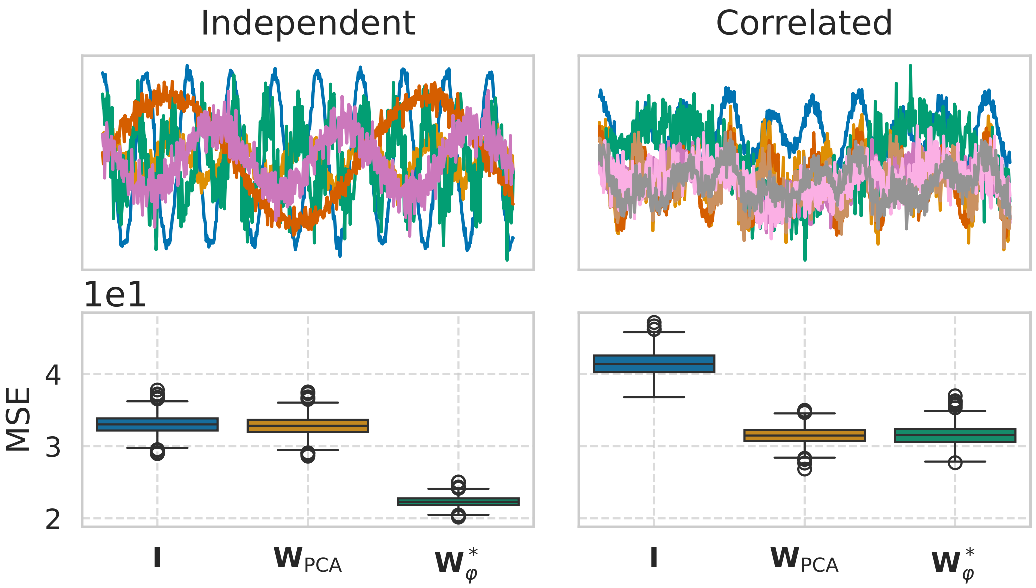


Figure 5: **Optimality of $\mathbf{W}^*_\varphi$.** Comparing the MSE obtained with $\mathbf{W}^*_\varphi$ against the baseline, for 1000 randomly generated linear FM.

Fig. 5 shows that in the case of uncorrelated data (*left* column) PCA is equivalent to the identity matrix, while the solution $\mathbf{W}^*_\varphi$ to the problem $\mathcal{L}(\mathbf{W}_\varphi)$ reaches an order of magnitude better forecasting loss. In the correlated case, we observe that PCA has a similar performance to the optimal solution. This example motivates the adapter idea through the existence of better linear transformations than the identity matrix in the case of linear foundation models.

B.3 Probabilistic adapters

We now present an alternative to the optimization of adapters, which is based on a Bayesian treatment of their parameters. There are many options on how to carry out inference over these parameters, and we can draw from the literature on Bayesian inference for neural networks (Papamarkou et al., 2024).

Considering a FM which yields point predictions, the appeal of Bayesian adapters is that they enable probabilistic predictions, which can be used for uncertainty quantification. Note that this is the case for models such as `Chronos` and `Moirai`, which output a distribution over the time series continuous values[3]. For deterministic FMs such as `Moment` (Goswami et al., 2024), a Bayesian treatment of adapters yields an ensemble of such predictions, which is key for accounting for the predictive uncertainty.

Inference. Recalling that θ represents the set of parameters of encoder (enc_θ) and decoder (dec_θ), we can attempt to obtain the posterior distribution over these parameters through Bayes theorem Gelman et al. (2013):

$$p(\theta|\mathbf{Y}, \mathbf{X}) \propto p(\mathbf{Y}|\mathbf{X}, \theta)p(\theta),$$

where $p(\theta)$ is the prior distribution over the parameters and $p(\mathbf{Y}|\mathbf{X}, \theta)$ the likelihood, with $\mathbf{Y}, \mathbf{X}$ representing a training dataset in this context. Alternatively, we can rather treat the latent representation $\mathbf{Z}$ as stochastic, where the interest is now to characterize the following posterior:

$$p(\mathbf{Z}|\mathbf{Y}, \mathbf{X}) \propto p(\mathbf{Y}|\mathbf{X}, \mathbf{Z})p(\mathbf{Z}).$$

In these two formulations, the posterior distribution over the parameters, is instrumental in obtaining predictive distributions useful for uncertainty quantification. For instance, in the case of inference over θ, for new test data $\mathbf{Y}^*, \mathbf{X}^*$ we obtain:

$$p(\mathbf{Y}^*|\mathbf{X}^*, \mathbf{Y}, \mathbf{X}) = \int p(\mathbf{Y}^*|\mathbf{X}^*, \theta)p(\theta|\mathbf{Y}, \mathbf{X})d\theta.$$

Characterizing the posterior analytically, however, is intractable and we need to resort to approximations. The literature on Bayesian inference offers various strategies, which can be adapted to neural networks (Papamarkou et al., 2024), including variational inference (Graves, 2011), Laplace approximations (Yang et al., 2024), and Markov chain Monte Carlo (MCMC) (Chen et al., 2014; Tran et al., 2022).

Within the **`AdaPTS`** framework, we focus in particular on variational inference (VI) for `VAE` adapters and on Monte Carlo dropout (Gal & Ghahramani, 2016) as an approximate form of VI for carrying out inference over θ.

Treating adapters in a Bayesian manner while keeping the FM fixed aligns with the concept of partially stochastic Bayesian neural networks, which provides theoretical guarantees on universal conditional density estimation (Sharma et al., 2023). This framework ensures that the model can approximate any conditional density, provided that stochasticity is introduced early enough in the architecture and that the number of stochastic units matches or exceeds the output dimension. Using probabilistic adapters, We comply with these conditions by making the encoder stochastic, allowing the learned latent space to capture uncertainty while leveraging the FM's fixed parameters.

B.4 Proof of Proposition 2.2

To derive the evidence lower bound (ELBO) used in the training objective of the VAE adapter, we start from the marginal likelihood of the observed data $\mathbf{Y}$ given the inputs $\mathbf{X}$ and foundation model f_{FM}. The marginal likelihood is expressed as:

$$\log p_\theta(\mathbf{Y}|\mathbf{X}, f_{\text{FM}}) = \log \int p_\theta(\mathbf{Y}|\mathbf{X}, f_{\text{FM}}(\mathbf{Z}))p(\mathbf{Z})\, d\mathbf{Z}, \tag{5}$$

where $\mathbf{Z}$ is the latent variable, $p_\theta(\mathbf{Y}|\mathbf{X}, f_{\text{FM}}(\mathbf{Z}))$ is the likelihood model parameterized by θ, and $p(\mathbf{Z})$ is the prior distribution over the latent variable $\mathbf{Z}$.

[3] In the case of `Chronos`, this distribution is obtained through a categorical distribution (with *softmax* probabilities) over a tokenized space of the time series values.

Direct optimization of this marginal likelihood is generally intractable due to the integration over $\mathbf{Z}$. To make this optimization feasible, we introduce a variational distribution $q_\phi(\mathbf{Z}|\mathbf{X})$, parameterized by ϕ, as an approximation to the true posterior $p_\theta(\mathbf{Z}|\mathbf{X}, \mathbf{Y}, f_{\mathrm{FM}})$. Using $q_\phi(\mathbf{Z}|\mathbf{X})$, we can reformulate the log-marginal likelihood as follows:

$$\log p_\theta(\mathbf{Y}|\mathbf{X}, f_{\mathrm{FM}}) = \log \int q_\phi(\mathbf{Z}|\mathbf{X}) \frac{p_\theta(\mathbf{Y}|\mathbf{X}, f_{\mathrm{FM}}(\mathbf{Z}))p(\mathbf{Z})}{q_\phi(\mathbf{Z}|\mathbf{X})} \, d\mathbf{Z} \tag{6}$$

$$= \log \mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X})} \left[\frac{p_\theta(\mathbf{Y}|\mathbf{X}, f_{\mathrm{FM}}(\mathbf{Z}))p(\mathbf{Z})}{q_\phi(\mathbf{Z}|\mathbf{X})} \right]. \tag{7}$$

Using Jensen's inequality, we can derive a lower bound on this log-marginal likelihood:

$$\log p_\theta(\mathbf{Y}|\mathbf{X}, f_{\mathrm{FM}}) \geq \mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X})} \left[\log \frac{p_\theta(\mathbf{Y}|\mathbf{X}, f_{\mathrm{FM}}(\mathbf{Z}))p(\mathbf{Z})}{q_\phi(\mathbf{Z}|\mathbf{X})} \right] \tag{8}$$

$$= \mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X})} \left[\log p_\theta(\mathbf{Y}|\mathbf{X}, f_{\mathrm{FM}}(\mathbf{Z})) \right] - \mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X})} \left[\log \frac{q_\phi(\mathbf{Z}|\mathbf{X})}{p(\mathbf{Z})} \right]. \tag{9}$$

The second term can be rewritten as the Kullback-Leibler (KL) divergence between the variational posterior $q_\phi(\mathbf{Z}|\mathbf{X})$ and the prior $p(\mathbf{Z})$:

$$\mathrm{KL}\left(q_\phi(\mathbf{Z}|\mathbf{X}) \,\|\, p(\mathbf{Z})\right) = \mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X})} \left[\log \frac{q_\phi(\mathbf{Z}|\mathbf{X})}{p(\mathbf{Z})} \right]. \tag{10}$$

Substituting this into the inequality, we obtain the evidence lower bound (ELBO):

$$\log p_\theta(\mathbf{Y}|\mathbf{X}, f_{\mathrm{FM}}) \geq \mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X})} \left[\log p_\theta(\mathbf{Y}|\mathbf{X}, f_{\mathrm{FM}}(\mathbf{Z})) \right] - \mathrm{KL}\left(q_\phi(\mathbf{Z}|\mathbf{X}) \,\|\, p(\mathbf{Z})\right). \tag{11}$$

The ELBO consists of two terms:

- The *forecasting* term, $\mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X})} \left[\log p_\theta(\mathbf{Y}|\mathbf{X}, f_{\mathrm{FM}}(\mathbf{Z})) \right]$, which measures how well the model can reconstruct $\mathbf{Y}$ given the latent variable $\mathbf{Z}$.
- The *regularization* term, $\mathrm{KL}\left(q_\phi(\mathbf{Z}|\mathbf{X}) \,\|\, p(\mathbf{Z})\right)$, which encourages the variational posterior to stay close to the prior distribution $p(\mathbf{Z})$.

Thus, maximizing the ELBO provides a tractable way to train the parameters θ and ϕ by optimizing the balance between forecasting accuracy and latent space regularization. □

C Normalizing Flows

Normalizing Flows make use of invertible transformations to map a simple base distribution (e.g. Gaussian) to a complex data distribution. Each transformation T is designed to maintain invertibility and efficient Jacobian computation. The transformation is applied iteratively: $\mathbf{Z} = T_k \circ T_{k-1} \circ \cdots \circ T_1(\mathbf{X})$. Current Normalizing Flow instantiations (e.g. `RealNVP`) make use of generic invertible transformations such as *coupling flows*; the latters can be parametrized using a neural network leading to powerful non-linear generative models that are trained to maximize the data log-likelihood:

$$\log p(\mathbf{X}) = \log p(\mathbf{Z}) + \sum_{i=1}^{k} \log \left| \det \frac{\partial T_i(\cdot;\theta)}{\partial \mathbf{Z}_{i-1}} \right|$$

where θ denote the parameters of the non-linear parametrization of the invertible transformations T_i, and $\mathbf{Z}_{i-1}$ is the output of the transformation T_{i-1}. In the context of time series adapters, we directly optimize the parameters of the transformations based on their direct and inverse application on the time series forecasting problem:

$$\mathcal{L}_{\text{flow}} = \|\mathbf{Y} - T_1^{-1} \circ T_2^{-1} \circ \cdots \circ T_k^{-1}(\\ f_{\mathrm{FM}}\big(T_k \circ T_{k-1} \circ \cdots \circ T_1(\mathbf{X};\theta)\big);\theta)\|_F^2$$

where the encoder is represented by the series of direct transformations: $\mathrm{enc}(\cdot) = T_k \circ T_{k-1} \circ \cdots \circ T_1(\cdot;\theta)$, and respectively the decoder by the series of inverse transformations $\mathrm{dec}(\cdot) = T_1^{-1} \circ T_2^{-1} \circ \cdots \circ T_k^{-1}(\cdot;\theta)$.

As defined here, Normalizing Flows suffer from the constraint of keeping the same dimension in both original and learned representation space. For this purpose, we investigate coupling a normalizing flow with a linear encoder-decoder type of architecture to enable dimensionality reduction prior to applying the transformation T_i. The parameters of the additional encoder and decoder are then jointly trained to optimize the learning objective $\mathcal{L}_{\text{flow}}$.

Given that the parameters of the encoder and the decoder are shared in Normalizing Flows, the gradient-based optimization within our framework receives conflicting directions due to gradient flow from both the direct and inverse transformations simultaneously. We discovered that this adapter construction was challenging to optimize in practice, and we defer the exploration of this direction to future research endeavors.

D Experimental setup

D.1 Datasets

Our experiments are conducted on four publicly available real-world multivariate time series datasets, commonly used for long-term forecasting (Ilbert et al., 2024; Wu et al., 2021; Chen et al., 2023; Nie et al., 2023; Zeng et al., 2023). These datasets include the Electricity Transformer Temperature dataset (ETTh1) (Zhou et al., 2021), ExchangeRate (Lai et al., 2018), Weather (Max Planck Institute, 2021), and Influenza-like Illness (U.S. Centers for Disease Control and Prevention, 2024). All time series are segmented with an input length of $L = 512$, prediction horizons $H \in [96, 192]$ and $H \in [24, 60]$ for the Illness dataset, and a stride of 1, meaning each subsequent window is shifted by one step. These datasets originate from various application domains, enabling a comprehensive evaluation of our framework across diverse real-world scenarios.

Table 2: Characteristics of the multivariate time series datasets used in our experiments with various sizes and dimensions.

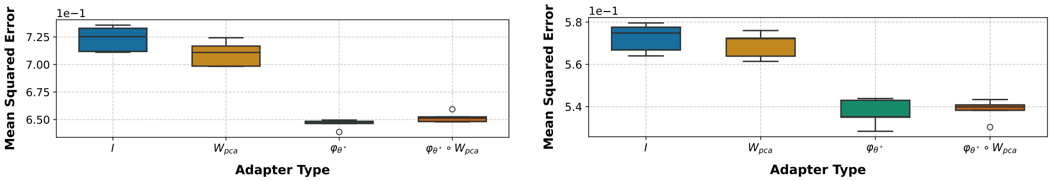
Dataset	ETTh1	Illness	ExchangeRate	Weather
# features	7	7	8	21
# time steps	13603	169	6791	51899
Granularity	1 hour	1 week	1 day	10 minutes
(Train, Val, Test)	(8033, 2785, 2785)	(69, 2, 98)	(4704, 665, 1422)	(36280, 5175, 10444)

D.2 Implementation details

In this section, we describe the full **`AdaPTS`** framework, starting from the data preprocessing, the training algorithm, and the hyperparameters optimization.

Preprocessing. Given that the adapter as defined in Definition 2.1 is a feature space transformation, we start by rescaling (*StandardScaler* and *MinMaxScaler*) the data where all the timesteps are regarded as data points. To account for the temporal specificities in each batch, we use Reversible Instance Normalization (RevIn) (Kim et al., 2022) that has been proven to mitigate time-related distribution shifts in time series problems.

Training parameters. After the pre-processing phase, we proceed to split the data into a *train-validation-test* sets, where the validation set serves as a tool to select the best hyperparameters for the adapter. The resulting adapter that is instantiated with the optimal hyperparameters is then tested against the unseen test dataset. For all of our experiments, we first train the linear forecasting head of `Moment` (referred to as *Linear Probing* in Goswami et al. (2024)) with the Adam optimizer (Kingma & Ba, 2017), a batch size of 32, a one cycle scheduler starting with 0.001 as learning rate. Once the forecasting linear head is trained, we freeze its parameters and proceed to training the adapter. This is



(a) Independent

(b) Correlated

Figure 6: `Moment` on simulated independent data.

done using the Adam optimizer, a batch size of 32, a reduce on plateau scheduler starting with 0.001 as learning rate.

Hyperparameter optimization. In order to select the best hyperparameters for the adapter architecture we use *Ray tune* (Liaw et al., 2018) with the Heteroscedastic and Evolutionary Bayesian Optimisation solver (HEBO) (Cowen-Rivers et al., 2022) engine, reporting the average mean squred error (MSE) from *k-fold* cross validation. Table 3 shows the default hyperparameters for each considered adapter.

Table 3: Adapters hyperparameters.

adapter	`LinearAE`	`DropoutLinearAE`	`LinearVAE`	`VAE`
p dropout	–	0.1	–	–
Number of layers	–	–	–	2
Hidden dimension	–	–	–	128
β	–	–	0.5	0.5
σ	–	–	1.0	1.0

E Additional results

E.1 `Moment` applied to synthetic data.

To validate the adapter *optimality* condition with large non-linear foundation models, we use `Moment` (Goswami et al., 2024). The optimal linear adapter in this case minimizes the following intractable objective:

$$\mathcal{L}(\mathbf{W}_{\varphi}) = \|\mathbf{Y} - f_{\texttt{Moment}}\big(\mathbf{X}\mathbf{W}_{\varphi}\big)\mathbf{W}_{\varphi}^{-1}\|_{\mathrm{F}}^{2} \tag{12}$$

To approximately solve this optimization problem, we instantiate $\mathbf{W}_{\varphi}$ as a single-linear-layer encoder denoted enc_{θ}, and respectively the inverse transformation $\mathbf{W}_{\varphi}^{-1}$ as a single-linear-layer decoder denoted dec_{θ}. We then use gradient-based optimization of the parameters θ using the Adam optimizer, aiming at solving the following optimization problem:

$$\theta^{*} = \arg\min_{\theta} \|\mathbf{Y} - \mathrm{dec}_{\theta}\big(f_{\texttt{Moment}}(\mathrm{enc}_{\theta}(\mathbf{X}))\big)\|_{\mathrm{F}}^{2} \tag{13}$$

Fig. 6 shows the performance gain obtained by optimizing a linear adapter on *`Moment`-small* foundation model. Unlike the tractable case, we observe that in both data modalities (independent and correlated data), `PCA` has little to no improvement over the identity baseline, while $\varphi_{\theta*}$ reaches an order of magnitude better solution. This confirms our intuition about the existence of a better solution than the identity matrix, even in the case of real-world complex foundation models.

E.2 Mean Absolute Error

Table 4 shows the comparison of our method with baselines in terms of the Mean Absolute Error (MAE).

Dataset	H	No adapter	with adapter				
		Moment$_{\text{small}}$	pca	linear	dropout	linear VAE	VAE
ETTh1	96	$0.422_{\pm 0.006}$	$0.440_{\pm 0.000}$	$0.423_{\pm 0.003}$	$\mathbf{0.415_{\pm 0.002}}$	$0.420_{\pm 0.001}$	$0.426_{\pm 0.001}$
	192	$\mathbf{0.436_{\pm 0.000}}$	$0.445_{\pm 0.000}$	$0.449_{\pm 0.003}$	$0.450_{\pm 0.001}$	$0.451_{\pm 0.001}$	$0.444_{\pm 0.001}$
Illness	24	$1.143_{\pm 0.007}$	$1.163_{\pm 0.001}$	$2.624_{\pm 0.035}$	$1.156_{\pm 0.016}$	$1.074_{\pm 0.011}$	$\mathbf{1.057_{\pm 0.012}}$
	60	$1.149_{\pm 0.001}$	$1.161_{\pm 0.001}$	$1.227_{\pm 0.030}$	$1.173_{\pm 0.015}$	$1.112_{\pm 0.021}$	$\mathbf{1.105_{\pm 0.021}}$
Weather	96	$0.232_{\pm 0.010}$	$0.235_{\pm 0.000}$	$0.226_{\pm 0.000}$	$0.212_{\pm 0.001}$	$\mathbf{0.218_{\pm 0.001}}$	$0.243_{\pm 0.001}$
	192	$\mathbf{0.251_{\pm 0.001}}$	$0.260_{\pm 0.001}$	$\mathbf{0.251_{\pm 0.001}}$	$\mathbf{0.251_{\pm 0.000}}$	$0.255_{\pm 0.000}$	$0.274_{\pm 0.000}$
ExchangeRate	96	$\mathbf{0.252_{\pm 0.010}}$	$0.264_{\pm 0.000}$	$0.308_{\pm 0.010}$	$0.269_{\pm 0.012}$	$0.376_{\pm 0.031}$	$0.488_{\pm 0.003}$
	192	$\mathbf{0.329_{\pm 0.001}}$	$0.335_{\pm 0.000}$	$0.415_{\pm 0.002}$	$0.419_{\pm 0.010}$	$0.513_{\pm 0.010}$	$0.585_{\pm 0.008}$

Table 4: Performance comparison between the baseline `Moment` model without adapters against different adapter architectures (`PCA`, `LinearAE`, `dropoutLAE`, `LinearVAE`, and `VAE`), for multivariate long-term forecasting with different horizons H. We display the average test MAE ± standard error obtained on 3 runs with different seeds. **Best** results are in bold, with lower values indicating better performance.

E.3 Ablation studies

Influence of σ and β in the `VAE` Adapter. Fig. 7 illustrates an ablation study examining the β parameter in β-`VAE` and the noise scale σ of the likelihood model applied to the prediction $\hat{\mathbf{Y}}$, assessing their effects on MSE and Expected Calibration Error (ECE). The MSE heatmap (left) demonstrates that increasing β generally diminishes MSE, with the lowest values observed at $\beta = 2.0$ and $\beta = 4.0$, particularly for higher $\log \sigma^2$. This indicates that stronger regularization through β can enhance forecasting accuracy, possibly due to the disentangling effect of regularization towards a prior distribution with statistically independent components. Conversely, the ECE heatmap (right) shows that higher β and $\log \sigma^2$ values result in lower calibration error, with optimal results at $\beta = 4.0$ and $\log \sigma^2 = 3.0$. This outcome is anticipated, as larger values of β and σ mitigate overfitting, where the model tends to exhibit overconfidence in its predictions. Additionally, it is observed that maintaining a fixed σ during training generally outperforms including it in the optimization loop, a configuration denoted as *auto* in Fig. 7.

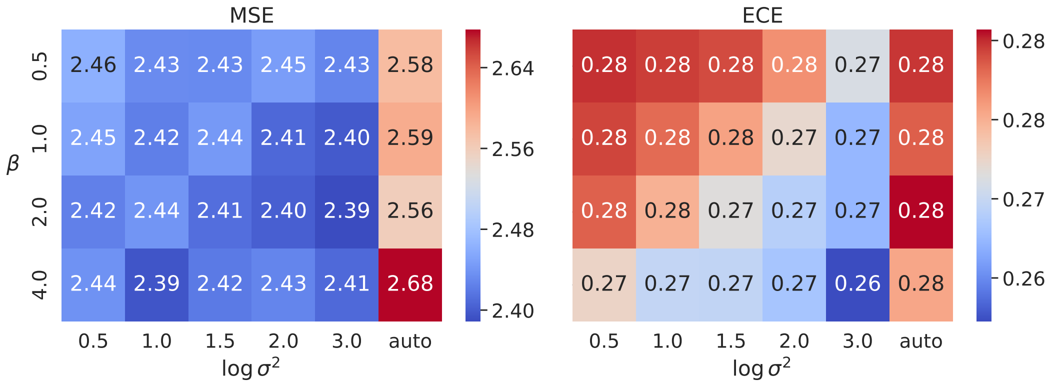


Figure 7: β and $\log \sigma^2$ `VAE` hyperparameters ablation on the Illness($H = 24$) dataset. For reference, the `Moment` baseline score on this task is $2.902_{\pm 0.023}$.

`LinearAE` components. The ablation study presented in Fig. 8 examines the performance of different components of the linear autoencoder adapter (`LinearAE`) across three datasets: ETTh1, Weather, and ExchangeRate. The figure compares the full linear autoencoder with its encoder-only (`LinearEncoder`) and decoder-only (`LinearDecoder`) variants. Overall, the results reveal that the decoder component of the linear autoencoder plays the most important role in minimizing the forecasting error across all datasets. The encoder-only variant's contribution varies, being more impactful in the Weather dataset compared to ETTh1 and ExchangeRate. These findings highlight the significance of the decoder in the `LinearAE` adapter and suggest that, in the deterministic case, a decoder might be sufficient to capture feature dependencies.

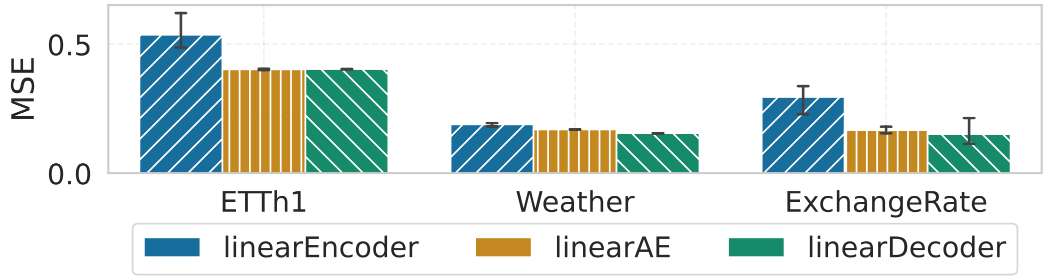


Figure 8: LinearAE components ablation.

Nevertheless, as shown in our previous experiments, particularly Table 1, probabilistic adapters generally outperformed the deterministic ones. This underscores the importance of the encoder as well, which is responsible for approximating the posterior distribution in the latent space—a mechanism inherent to our probabilistic framework.