Appendix

A Additional Experiments[3]

A.1 Experiments on the ETT datasets

In the main body, we present a comparison of the benchmark methods on the ETTm2 dataset. In this section, we extend our analysis to the remaining three ETT datasets, namely ETTh1, ETTh2, and ETTm1, as summarized in Table 7. Our experimental results reveal that Basisformer outperforms all other methods in terms of MSE and MAE. Specifically, Basisformer demonstrates a superior average MSE reduction of 1.32% , 6.74% and 9.23% when compared to FiLM, Fedformer and DLinear, respectively.

Table 7: Multivariate results for the remaining three ETT datasets using an input length of $I = 96$ (or $I = 36$ for the illness dataset) and output lengths of $O \in \{96, 192, 336, 720\}$ (or $O \in \{24, 36, 48, 60\}$ for the illness dataset). In all experiments, lower MSE values indicate better model performance, and we present the best results in boldface.

Models		Fedformer		Autoformer		N-HiTS		FiLM		Dlinear		Informer		Basisformer	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.376**	0.419	0.449	0.459	0.419	0.413	0.388	0.401	0.386	**0.400**	0.865	0.713	0.394	0.411
	192	**0.420**	0.448	0.500	0.482	0.468	0.443	0.443	0.439	0.437	**0.432**	1.008	0.792	0.442	0.437
	336	**0.459**	0.465	0.521	0.496	0.551	0.489	0.484	0.461	0.481	0.459	1.107	0.809	0.473	**0.451**
	720	0.506	0.507	0.514	0.512	0.669	0.559	0.525	0.519	0.519	0.516	1.181	0.865	**0.460**	**0.465**
ETTh2	96	0.358	0.397	0.346	0.388	0.374	0.383	**0.292**	**0.341**	0.333	0.387	3.755	1.525	0.312	0.356
	192	0.429	0.439	0.456	0.452	0.476	0.446	**0.378**	**0.396**	0.477	0.476	5.602	1.931	0.382	0.401
	336	0.496	0.487	0.482	0.486	0.472	0.446	0.426	0.438	0.594	0.541	4.721	1.835	**0.418**	**0.431**
	720	0.463	0.474	0.515	0.511	0.932	0.636	0.443	0.455	0.831	0.657	3.647	1.625	**0.418**	**0.438**
ETTm1	96	0.379	0.419	0.505	0.475	**0.324**	**0.349**	0.357	0.373	0.345	0.372	0.672	0.571	0.342	0.374
	192	0.426	0.441	0.553	0.496	**0.376**	**0.379**	0.387	0.385	0.380	0.389	0.795	0.669	0.380	0.392
	336	0.445	0.459	0.621	0.537	**0.409**	**0.405**	0.420	0.407	0.413	0.413	1.212	0.871	0.420	0.418
	720	0.543	0.490	0.671	0.561	**0.472**	0.443	0.478	**0.439**	0.474	0.453	1.166	0.823	0.492	0.458

A.2 Experimental results with longer length input setting

Throughout our research, we maintain consistency in our experimental settings by fixing the input length to be 96 (with a reduced input length of 36 for the illness dataset), instead of using a longer length. The main rationale behind this decision is that, in practical scenarios where the model is deployed as an online service and tasked with predicting a long range of the future at a granular level of minutes or hours, collecting a lengthy history (i.e., spanning 720 timestamps) for a large number of time series in real-time can be quite challenging. Therefore, the adoption of an input length of 96 proves to be more practical and feasible.

Given that certain recent methods utilize longer input lengths to yield better performance, irrespective of the length, we present supplementary comparison outcomes with extended input lengths in Table 8. Specifically, Fedformer, Autoformer, and TCN exhibit a decline in performance with an increase in input length, and hence, we retain their original outcomes at an input length of 96. In contrast, Dlinear employs an input length of 336 (104 for the illness dataset) by default, FiLM utilizes an input length that is at most four times of the output length, and N-HiTS adopts an input length that is five times of the output length. To enable a fair comparison, we standardize our input length for longer inputs to 192 (72 for the illness dataset).

The experimental results yield several notable findings. Firstly, those methods that benefit from longer inputs, namely Dlinear, FiLM, and N-HiTS, exhibit a significant performance decline when the input length is reduced from longer settings to an input length of 96. Concretely, Dlinear, FiLM, and N-HiTS show performance declines of 25.82%, 19.48%, and 330.42%, respectively. Conversely, our approach maintains most of its performance with a slight deterioration of 6.23%, as evident in Table 1 and Table 8. Secondly, concerning longer inputs, our method surpasses recent approaches such as Dlinear, FiLM, and N-HiTS, with an average MSE performance improvement of 1.35%, 0.63%, and 7.75%, respectively, and a corresponding evaluation MAE performance improvement of 3.15%, 2.33%, and 4.06%, respectively. It is noteworthy that our approach requires an input length

[3]All the six datasets can be downloaded from `https://drive.google.com/drive/folders/1ZOYpTUa82_jCcxIdTmyr0LXQfvaM9vIy?usp=sharing`

Table 8: Multivariate results for six datasets using a longer input length. Lower MSE indicate superior model performance, and the best results are presented in boldface.

Models		Fedformer		Autoformer		N-HiTS		FiLM		Dlinear		TCN		Basisformer	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETT	96	0.203	0.287	0.255	0.339	0.176	**0.255**	**0.165**	0.256	0.167	0.260	3.041	1.330	0.185	0.270
	192	0.269	0.328	0.281	0.340	0.245	0.305	**0.222**	**0.296**	0.224	0.303	3.072	1.339	0.247	0.307
	336	0.325	0.366	0.339	0.372	0.295	0.346	**0.277**	**0.333**	0.281	0.342	3.105	1.348	0.298	0.341
	720	0.421	0.415	0.422	0.419	0.401	0.416	**0.371**	**0.389**	0.397	0.421	3.135	1.354	0.381	0.393
electricty	96	0.193	0.308	0.201	0.317	0.147	0.249	0.154	0.267	**0.140**	**0.237**	0.985	0.813	0.145	0.245
	192	0.201	0.315	0.222	0.334	0.167	0.269	0.164	0.258	**0.153**	**0.249**	0.996	0.821	0.165	0.263
	336	0.214	0.329	0.231	0.338	0.186	0.290	0.188	0.283	**0.169**	**0.267**	1.000	0.824	0.178	0.276
	720	0.246	0.355	0.254	0.361	0.243	0.340	0.236	0.332	**0.203**	**0.301**	1.438	0.784	0.219	0.310
exchange	96	0.148	0.278	0.197	0.323	0.092	0.211	**0.079**	0.204	0.081	**0.203**	3.004	1.432	0.084	0.205
	192	0.271	0.380	0.300	0.369	0.208	0.322	0.159	**0.292**	**0.157**	0.293	3.048	1.444	0.172	0.298
	336	0.460	0.500	0.509	0.524	0.341	0.422	**0.270**	**0.398**	0.305	0.414	3.113	1.459	0.303	0.403
	720	1.195	0.841	1.447	0.941	0.888	0.723	**0.536**	**0.574**	0.643	0.601	3.150	1.458	0.781	0.668
traffic	96	0.587	0.366	0.613	0.388	**0.402**	**0.282**	0.416	0.294	0.410	0.282	1.438	0.784	0.403	0.293
	192	0.604	0.373	0.616	0.382	0.420	0.297	**0.408**	**0.288**	0.423	0.287	1.463	0.794	0.421	0.301
	336	0.621	0.383	0.622	0.387	0.448	0.313	0.425	0.298	0.436	**0.296**	1.479	0.799	**0.418**	0.298
	720	0.626	0.382	0.660	0.408	0.539	0.353	0.520	0.353	0.466	0.315	1.499	0.804	**0.464**	**0.312**
weather	96	0.217	0.296	0.266	0.336	**0.158**	**0.195**	0.199	0.262	0.176	0.237	0.615	0.589	0.168	0.215
	192	0.276	0.336	0.307	0.367	**0.211**	**0.247**	0.228	0.288	0.220	0.282	0.629	0.600	0.213	0.257
	336	0.339	0.380	0.359	0.395	0.274	0.300	0.267	0.323	0.265	0.319	0.639	0.608	**0.263**	**0.292**
	720	0.403	0.428	0.419	0.428	0.351	0.353	**0.319**	0.361	0.323	0.362	0.639	0.610	0.343	**0.346**
illness	24	3.228	1.260	3.486	1.287	1.862	0.869	1.970	0.875	2.215	1.081	6.624	1.830	**1.427**	**0.778**
	36	2.679	1.080	3.103	1.148	2.071	0.969	1.982	0.859	1.936	0.963	6.858	1.879	**1.464**	**0.813**
	48	2.622	1.078	2.669	1.085	2.184	0.999	1.868	0.896	2.130	1.024	6.968	1.892	**1.660**	**0.862**
	60	2.857	1.157	2.770	1.125	2.507	1.060	2.057	0.929	2.368	1.096	7.127	1.918	**1.853**	**0.917**

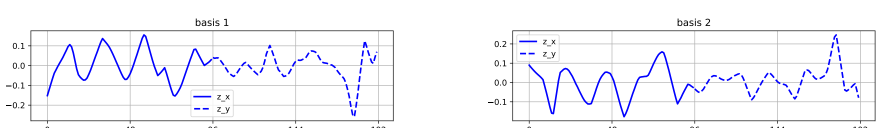
of 192 (72 for the illness dataset), which is at least 40% lower than the input length of the other three methods. Furthermore, for even longer input lengths, our model's performance can be further enhanced, signifying that our approach can leverage limited data more efficiently.

A.3 Additional abalation study

Impact of the number of basis vectors: We present the performance of the proposed model under varying numbers of basis vectors N in Table 9, where N is set to 1, 5, 10, 15, and 20. The results demonstrate that the model's performance remains stable over a wide range of N, indicating its ability to adaptively adjust to the number of basis vectors. Notably, when N increases beyond a certain threshold, some of the basis vectors may become redundant. To further explore this, we visualize a subset of the learned basis vectors when $N = 20$ in Figure 3. Interestingly, we observe a high cosine similarity of -0.93 between two of the bases, suggesting that some basis vectors may not be necessary for accurate prediction. Thus, in practical applications, we set N to 10 for all datasets to reduce computational complexity without compromising performance.

Table 9: The impact of the number of bases N on the performance of the model. The electricity dataset is employed in this experiment. We present the best results in boldface.

basis number	1		5		10		15		20	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
96	0.173	0.269	0.171	0.265	**0.166**	**0.259**	0.168	0.263	0.168	0.263
192	0.183	0.277	0.178	0.270	**0.176**	0.270	0.176	0.269	0.176	**0.268**
336	0.196	0.289	0.192	0.284	**0.190**	**0.283**	0.192	0.285	0.193	0.285
720	0.231	0.317	0.229	0.314	**0.218**	**0.306**	0.220	0.308	0.224	0.311
avg	0.196	0.288	0.192	0.283	**0.187**	**0.279**	0.189	0.281	0.190	0.282



(a) basis vector 1 (b) basis vector 2

Figure 3: Two highly correlated basis vectors when the number of basis vectors N is large.

Impact of the number of the BCAB layers: The ablation study on the number of BCABs is shown in Table 10. The findings indicate that stacking a certain number of BCAB modules can enhance the performance of the model. However, exceeding a certain threshold can lead to overfitting, resulting in a decline in performance. Hence, we recommend the use of two layers of BCABs in practical experiments to achieve optimal performance without overfitting the model.

Table 10: The impact of the number of stacked BCAB on the performance of the model. The electricity dataset is employed in this experiment. We present the best results in boldface.

BCAB number	1		2		3		4	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
96	0.166	0.260	**0.166**	**0.259**	0.168	0.263	0.171	0.266
192	0.176	0.270	**0.176**	**0.268**	0.176	0.269	0.179	0.272
336	**0.187**	**0.280**	0.190	0.283	0.190	0.283	0.191	0.284
720	0.228	0.313	**0.218**	**0.306**	0.234	0.319	0.237	0.319
avg	0.189	0.281	**0.187**	**0.279**	0.192	0.283	0.194	0.285

Impact of the bottleneck in the forecast module: The performance of the proposed model under varying bottleneck settings is presented in Table 11. The results demonstrate that employing a bottleneck architecture with a width of 48 can significantly reduce the number of model parameters without degrading the performance significantly, as opposed to not using a bottleneck architecture.

Table 11: The impact of the MLP bottleneck in the forecast module. The electricity dataset is employed in this experiment. Setting the bottleneck dimension to 96 is equivalent to not using a bottleneck since the input length is 96. The best results are highlighted in bold. The second best is underlined.

bottleneck	96		48		32		24	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
96	**0.163**	**0.257**	$\underline{0.166}$	$\underline{0.259}$	0.172	0.267	0.172	0.269
192	**0.172**	**0.265**	$\underline{0.176}$	$\underline{0.268}$	0.182	0.273	0.186	0.279
336	**0.186**	**0.279**	$\underline{0.190}$	$\underline{0.283}$	0.194	0.286	0.197	0.289
720	**0.217**	**0.305**	$\underline{0.218}$	$\underline{0.306}$	0.230	0.316	0.233	0.317
avg	**0.184**	**0.276**	$\underline{0.187}$	$\underline{0.279}$	0.195	0.286	0.197	0.288

A.4 Sensitivity analysis of the weights for the losses in Eq.(9)

Our model utilizes three distinct loss functions: the supervised MSE loss for prediction L_{pred}, the self-supervised InfoNCE loss for basis learning L_{align}, and the smoothness loss for smoothing the basis over time L_{align}. During training, we directly combine these loss functions as the model's performance is not significantly impacted by the relative weights of the individual losses within a certain range. This assertion is supported by the performance evaluation presented in Figure 4, which investigates the impact of different weight combinations of the three loss functions. In our setting, we fix the weight of the predicted loss function to be 1, and then fix either the weight of the contrast loss function or the smoothness loss function to be 1, while the other one varies within the range of $\{0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6\}$. To explore the inflection point of the effect, we take the middle point of two points and calculate a finer range again. Our results indicate that the contrast loss function is essentially stable between the weight range of 0.6-1.2, while the smoothness loss function is similarly stable between the weight range of 0.9-1.5.

A.5 Uncertainty of the results

To assess the stability of our proposed method, we performed 5 repeated experiments and calculated the standard deviations for all methods, as presented in Table 12. Notably, our method exhibits a relatively small variance within the table, indicating its high degree of stability.

B Implementation Details

The training and testing of BasisFormer are conducted on an NVIDIA GeForce RTX 3090 graphics card with 24268MB of VRAM. During the trainin process, we adopt the Adabelief optimizer [23] for

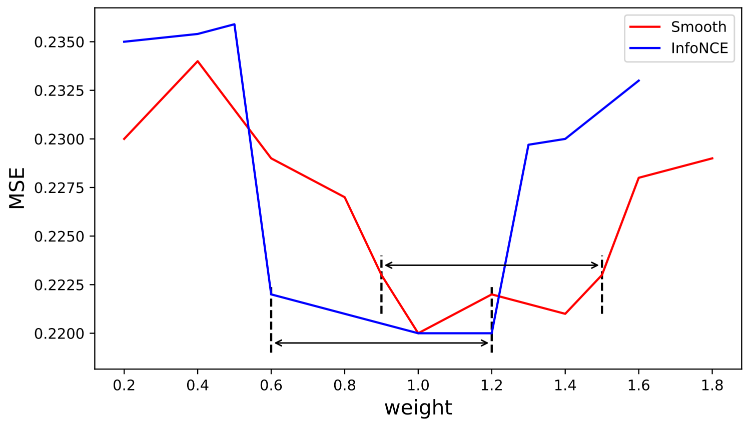


Figure 4: MSE for the testing data as a function of the weight for the smoothness (the red line) and the infoNCE loss(the blue line).

Table 12: Results for 6 benchmark datasets with standard deviations in the brackets.

Models		Fedformer		Autoformer		N-HiTS		FiLM		Dlinear		TCN		Basisformer	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETT	96	0.203 (0.002)	0.287 (0.001)	0.255 (0.020)	0.339 (0.020)	0.192 (0.003)	0.265 (0.002)	**0.183** (0.000)	0.266 (0.000)	0.193 (0.004)	0.292 (0.006)	3.041 (0.000)	1.330 (0.000)	0.184 (0.002)	**0.266** (0.002)
	192	0.269 (0.006)	0.328 (0.005)	0.281 (0.027)	0.340 (0.025)	0.287 (0.004)	0.329 (0.001)	**0.247** (0.000)	**0.305** (0.000)	0.284 (0.016)	0.362 (0.016)	3.072 (0.002)	1.339 (0.001)	0.248 (0.004)	0.307 (0.002)
	336	0.325 (0.002)	0.366 (0.003)	0.339 (0.018)	0.372 (0.015)	0.389 (0.005)	0.389 (0.003)	**0.309** (0.000)	**0.343** (0.000)	0.369 (0.006)	0.554 (0.002)	3.105 (0.005)	1.348 (0.003)	0.321 (0.005)	0.355 (0.004)
	720	0.421 (0.018)	0.415 (0.012)	0.422 (0.015)	0.419 (0.010)	0.591 (0.011)	0.491 (0.002)	**0.407** (0.001)	**0.399** (0.000)	0.554 (0.037)	0.522 (0.026)	3.135 (0.021)	1.354 (0.005)	0.410 (0.007)	0.404 (0.004)
electricity	96	0.193 (0.001)	0.308 (0.001)	0.201 (0.003)	0.317 (0.004)	1.748 (0.003)	1.020 (0.001)	0.199 (0.000)	0.276 (0.001)	0.199 (0.000)	0.284 (0.000)	0.985 (0.006)	0.813 (0.004)	**0.165** (0.001)	**0.259** (0.001)
	192	0.201 (0.005)	0.315 (0.006)	0.222 (0.003)	0.334 (0.004)	1.743 (0.008)	1.018 (0.003)	0.198 (0.000)	0.279 (0.001)	0.198 (0.000)	0.287 (0.000)	0.996 (0.008)	0.821 (0.007)	**0.178** (0.001)	**0.272** (0.001)
	336	0.214 (0.001)	0.329 (0.002)	0.231 (0.006)	0.338 (0.004)	1.677 (0.010)	1.000 (0.003)	0.217 (0.001)	0.301 (0.001)	0.210 (0.001)	0.302 (0.001)	1.000 (0.004)	0.824 (0.003)	**0.189** (0.001)	**0.282** (0.001)
	720	0.246 (0.003)	0.355 (0.003)	0.254 (0.007)	0.361 (0.008)	- -	- -	0.280 (0.000)	0.358 (0.000)	0.245 (0.000)	0.335 (0.000)	1.438 (0.006)	0.784 (0.003)	**0.223** (0.002)	**0.311** (0.001)
exchange	96	0.148 (0.004)	0.278 (0.004)	0.197 (0.019)	0.323 (0.012)	1.685 (0.042)	1.049 (0.017)	**0.083** (0.003)	**0.201** (0.003)	0.088 (0.004)	0.218 (0.005)	3.004 (0.128)	1.432 (0.070)	0.085 (0.004)	0.205 (0.005)
	192	0.271 (0.012)	0.380 (0.010)	0.300 (0.020)	0.369 (0.016)	1.658 (0.015)	1.023 (0.006)	0.179 (0.003)	0.300 (0.002)	**0.176** (0.005)	0.315 (0.006)	3.048 (0.020)	1.444 (0.008)	0.177 (0.005)	**0.299** (0.005)
	336	0.460 (0.009)	0.500 (0.007)	0.509 (0.041)	0.524 (0.016)	1.566 (0.037)	0.988 (0.015)	0.337 (0.005)	**0.416** (0.003)	**0.313** (0.006)	0.427 (0.006)	3.113 (0.082)	1.459 (0.021)	0.336 (0.011)	0.421 (0.007)
	720	1.195 (0.042)	0.841 (0.017)	1.447 (0.084)	0.941 (0.028)	1.809 (0.052)	1.055 (0.018)	**0.642** (0.040)	**0.610** (0.029)	0.839 (0.027)	0.695 (0.012)	3.150 (0.237)	1.458 (0.063)	0.854 (0.024)	0.670 (0.011)
traffic	96	0.587 (0.010)	0.366 (0.008)	0.613 (0.028)	0.388 (0.012)	2.138 (0.016)	1.026 (0.006)	0.652 (0.001)	0.395 (0.003)	0.650 (0.001)	0.396 (0.001)	1.438 (0.001)	0.784 (0.001)	**0.444** (0.003)	**0.315** (0.003)
	192	0.604 (0.012)	0.373 (0.009)	0.616 (0.042)	0.382 (0.020)	2.101 (0.015)	1.015 (0.007)	0.605 (0.001)	0.371 (0.003)	0.605 (0.002)	0.378 (0.001)	1.463 (0.032)	0.794 (0.010)	**0.460** (0.004)	**0.316** (0.002)
	336	0.621 (0.008)	0.383 (0.008)	0.622 (0.009)	0.387 (0.003)	- -	- -	0.615 (0.001)	0.372 (0.001)	0.612 (0.003)	0.382 (0.004)	1.479 (0.003)	0.799 (0.002)	**0.471** (0.005)	**0.317** (0.004)
	720	0.626 (0.004)	0.382 (0.003)	0.660 (0.025)	0.408 (0.015)	- -	- -	0.692 (0.000)	0.428 (0.000)	0.645 (0.001)	0.394 (0.001)	1.499 (0.010)	0.804 (0.005)	**0.486** (0.005)	**0.318** (0.004)
weather	96	0.217 (0.018)	0.296 (0.019)	0.266 (0.007)	0.336 (0.006)	0.648 (0.001)	0.492 (0.000)	0.193 (0.002)	0.234 (0.001)	0.196 (0.001)	0.255 (0.003)	0.615 (0.002)	0.589 (0.002)	**0.173** (0.003)	**0.214** (0.003)
	192	0.276 (0.015)	0.336 (0.017)	0.307 (0.024)	0.367 (0.022)	0.616 (0.003)	0.479 (0.001)	0.238 (0.000)	0.270 (0.001)	0.237 (0.001)	0.296 (0.002)	0.629 (0.023)	0.600 (0.009)	**0.223** (0.002)	**0.257** (0.001)
	336	0.339 (0.014)	0.380 (0.015)	0.359 (0.035)	0.395 (0.031)	0.579 (0.002)	0.462 (0.001)	0.288 (0.001)	0.304 (0.000)	0.283 (0.002)	0.335 (0.004)	0.639 (0.050)	0.608 (0.017)	**0.278** (0.001)	**0.298** (0.000)
	720	0.403 (0.009)	0.428 (0.008)	0.419 (0.017)	0.428 (0.014)	0.541 (0.001)	0.447 (0.000)	0.358 (0.001)	0.350 (0.000)	0.343 (0.020)	0.383 (0.020)	0.639 (0.050)	0.610 (0.018)	**0.355** (0.001)	**0.347** (0.001)
illness	24	3.228 (0.020)	1.260 (0.009)	3.486 (0.107)	1.287 (0.018)	3.297 (0.007)	1.679 (0.000)	2.198 (0.138)	0.911 (0.058)	2.398 (0.065)	1.040 (0.032)	6.624 (0.550)	1.830 (0.094)	**1.550** (0.087)	**0.814** (0.024)
	36	2.679 (0.018)	1.080 (0.005)	3.103 (0.139)	1.148 (0.025)	2.379 (0.136)	1.441 (0.043)	2.267 (0.077)	0.926 (0.059)	2.646 (0.137)	1.088 (0.064)	6.858 (0.216)	1.879 (0.034)	**1.516** (0.130)	**0.819** (0.030)
	48	2.622 (0.010)	1.078 (0.002)	2.669 (0.151)	1.085 (0.037)	3.341 (0.092)	1.751 (0.030)	2.348 (0.115)	0.989 (0.037)	2.614 (0.140)	1.086 (0.049)	6.968 (0.032)	1.892 (0.008)	**1.877** (0.110)	**0.907** (0.032)
	60	2.857 (0.011)	1.157 (0.003)	2.770 (0.085)	1.125 (0.019)	2.278 (0.187)	1.493 (0.064)	2.508 (0.130)	1.038 (0.018)	2.804 (0.049)	1.146 (0.009)	7.127 (0.134)	1.918 (0.025)	**1.878** (0.098)	**0.902** (0.024)

Experiment with '-' means it reported an out-of-memory error on a computer with 128G memory.

Table 13: Comparison of computational complexity for different models.

Methods	TIME	MEMORY
Fedformer	$\mathcal{O}(O)$	$\mathcal{O}(O)$
Autoformer	$\mathcal{O}(O \log O)$	$\mathcal{O}(O \log O)$
N-HiTS	$\mathcal{O}(O(1-r^B)/(1-r)$	$\mathcal{O}(O(1-r^B)/(1-r)$
FiLM	$\mathcal{O}(O)$	$\mathcal{O}(O)$
Dlinear	$\mathcal{O}(O)$	$\mathcal{O}(O)$
TCN	$\mathcal{O}(O)$	$\mathcal{O}(O)$
LogTrans	$\mathcal{O}(O \log O)$	$\mathcal{O}(O^2)$
Reformer	$\mathcal{O}(O \log O)$	$\mathcal{O}(O \log O)$
Informer	$\mathcal{O}(O \log O)$	$\mathcal{O}(O \log O)$
Basisformer	$\mathcal{O}(O)$	$\mathcal{O}(O)$

optimization. We train the model for 30 epochs with the patience of 3 epochs. All experiments are averaged over 5 trials.

To implement the multi-head mechanism, we calculate the multi-head attention for each CAB separately, and then restore it to the original dimension through multiplication, concatenation, and a linear layer. In the last layer of the network, a mapping layer was utilized to map it to H heads, and the dot product outputs the final coefficients.

To promote the learning of bases and ensure consistency of time series across different dimensions, we normalized the time series during training and performed inverse normalization when outputting the results.

For the other models compared in the table, we utilized their original code and conducted experiments by only varying the input length.

C Analysis of the Limitations of BasisFormer

BasisFormer demonstrates proficiency in learning effective representations and capturing the relationship between bases and time series. However, this proficiency is contingent upon the multi-dimensional time series being on the same feature scale, which necessitates normalization of the time series during training and inverse normalization when outputting results. Despite this, the normalization and inverse normalization operations introduce changes to the original distribution of the time series, making it arduous to fit certain distributions. As such, future work could explore alternative approaches to training on datasets with considerably different feature scales, eliminating the need for normalization and inverse normalization. Possible avenues for investigation include identifying appropriate mathematical methods or neural network transformations to map data to a suitable and universal feature space.

D Relation to Meta-learning

From a meta-learning standpoint, the learnable basis in our model is tantamount to meta-knowledge for all time series within the same window. The coefficients, which are derived from the similarity between each time series and the foundation, represent distinctive knowledge for each time series. Consequently, our model can be perceived as a manifestation of meta-learning. Notwithstanding, we departed from conventional meta-learning approaches by forgoing a two-stage inner-outer loop optimization method, instead opting for an end-to-end training method.

E Analysis of the Model Complexity

Suppose that the input and output length in BasisFormer is I and O respectively when forecasting a single time series. Note that the time and space complexity of BasisFormer are of the same order. Therefore, we refer to both of them as complexity in the sequel.

With regards to the coef module, the complexity is primarily determined by the cross-attention mechanism. Within our approach, BCAB utilizes attention on the channel dimension, and we encode the time sequence dimension to a specified hidden length $D_c \ll O$ via a linear layer during computation. Consequently, the complexity of this module is $\mathcal{O}(N)$, where N is the number of bases - a fixed hyperparameter which is usually not large. In this step, we omit the number of BCAB stacks M, since M is also a fixed hyperparameter. As previously mentioned in Appendix A.3, to limit overfitting, M is typically set to 2.

The prediction module incorporates two Multilayer Perceptron (MLP) networks, which are employed for separating and concatenating different heads. Both MLP networks have bottlenecks with constant values, and they carry a complexity of $\mathcal{O}(O)$. In terms of the aggregation of different base vectors, the complexity also is $\mathcal{O}(O)$. Therefore, the cumulative complexity of this module is $\mathcal{O}(O)$.

In summary, the total complexity of our model is $\mathcal{O}(O)$. Table 13 provides a comparison of the computational complexity among different models, and BasisFormer achieves the lowest complexity among them.