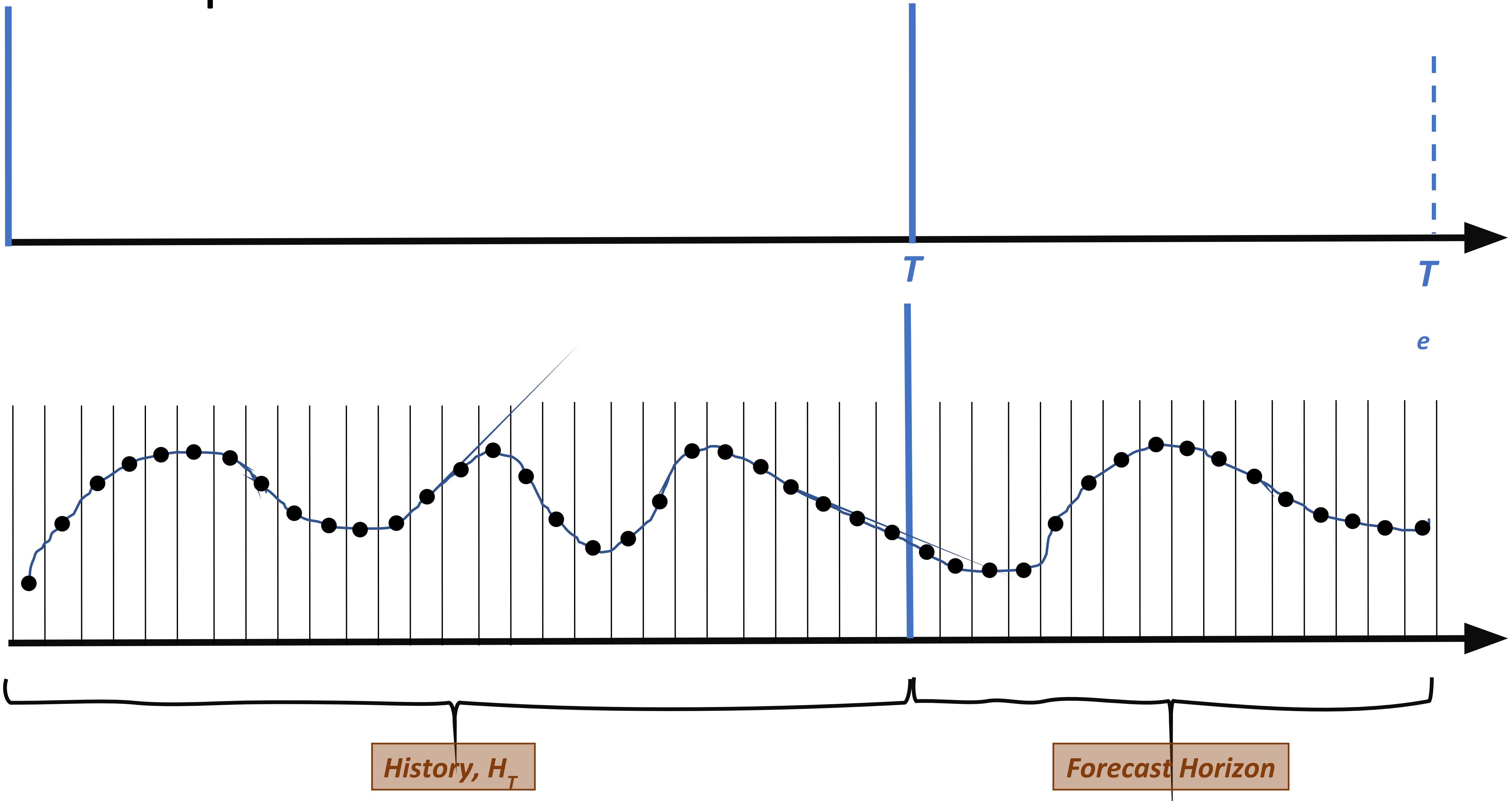
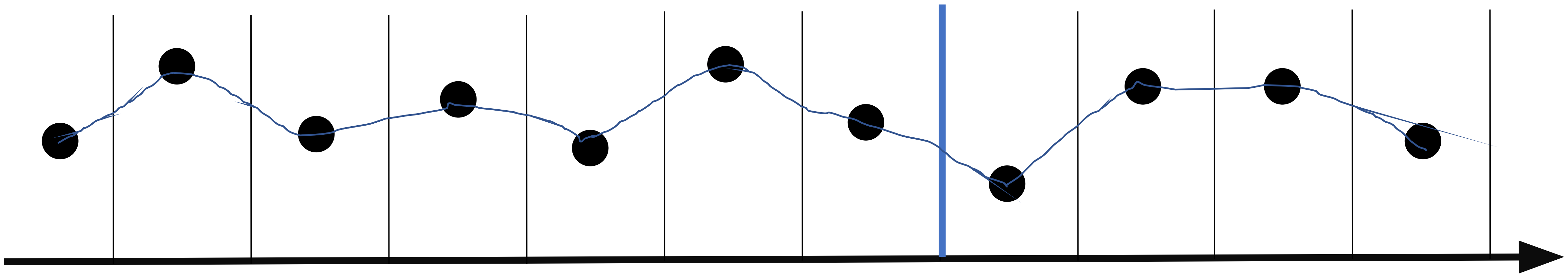
Coherent Probabilistic Aggregate Queries on Long-horizon Forecasts

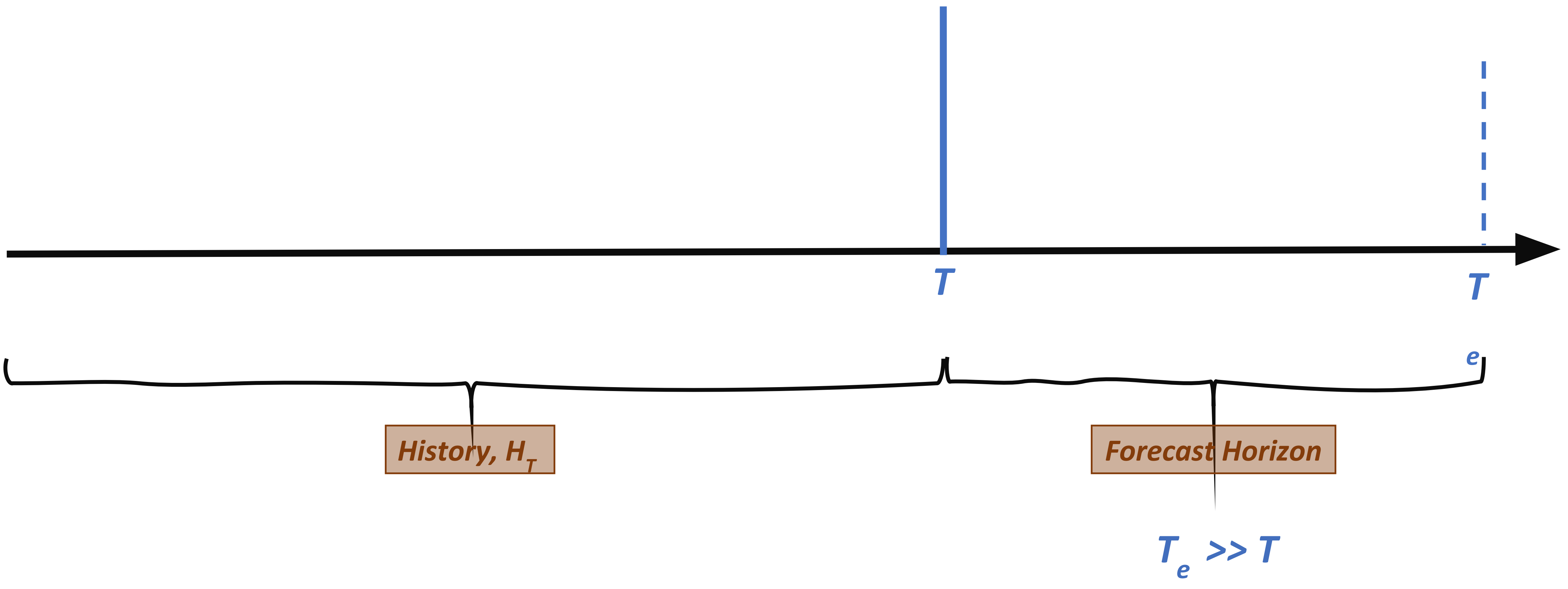
Shapes



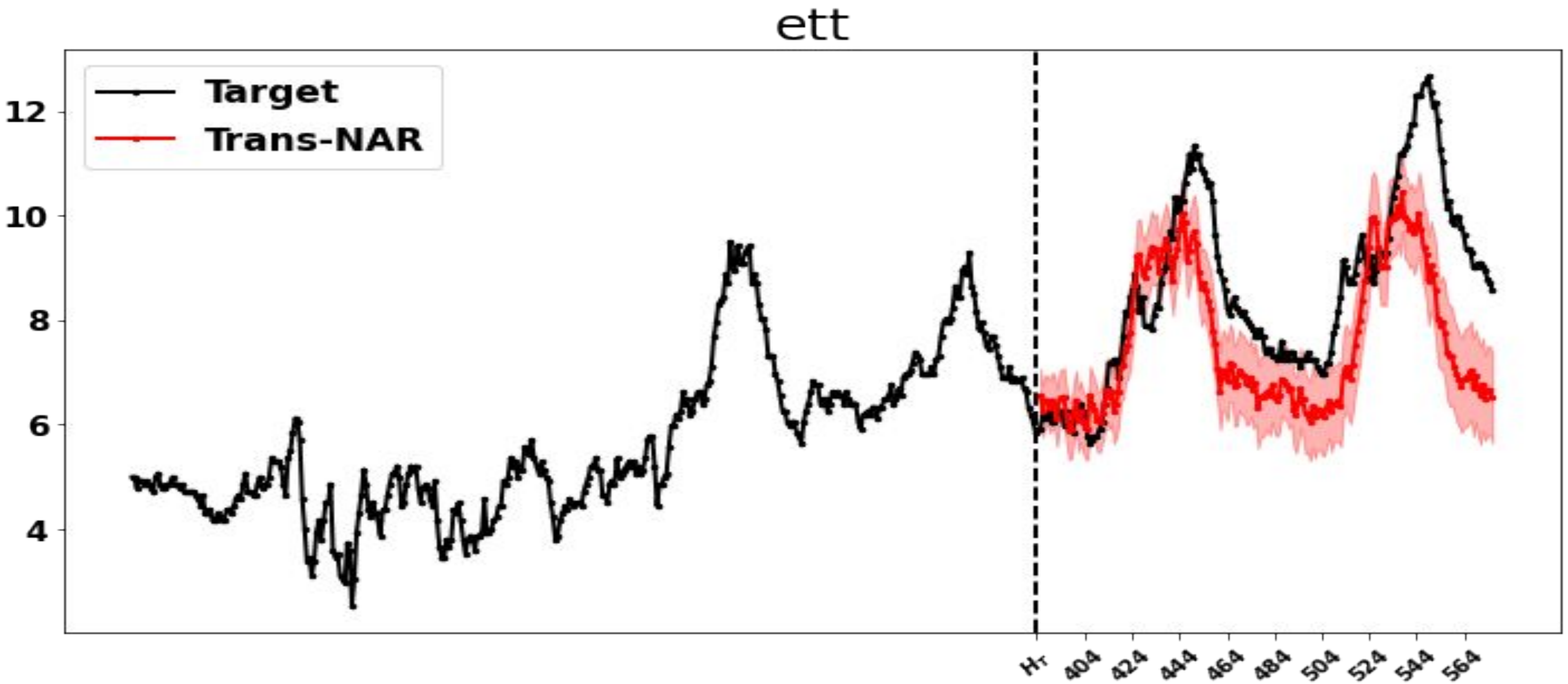


Probabilistic Forecasting

Goal: Predict the distribution at each position in the forecast horizon



Probabilistic Forecasting in Time-Series



- Red curve denotes the mean forecasts
- Shaded region around mean denotes two standard deviations confidence interval

Long-Range Forecasting

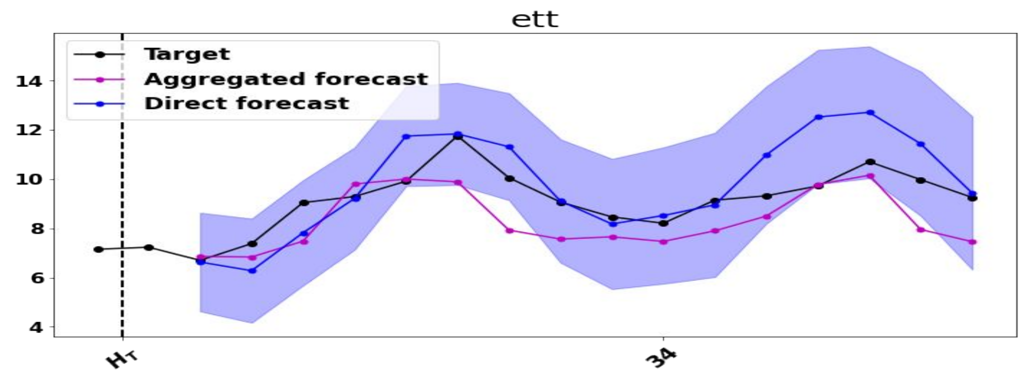
- Short-Term Forecasting -- typical forecast horizon is of few tens of values or less
- Long-Range Forecasting – Forecast horizon of few hundreds or thousands of values

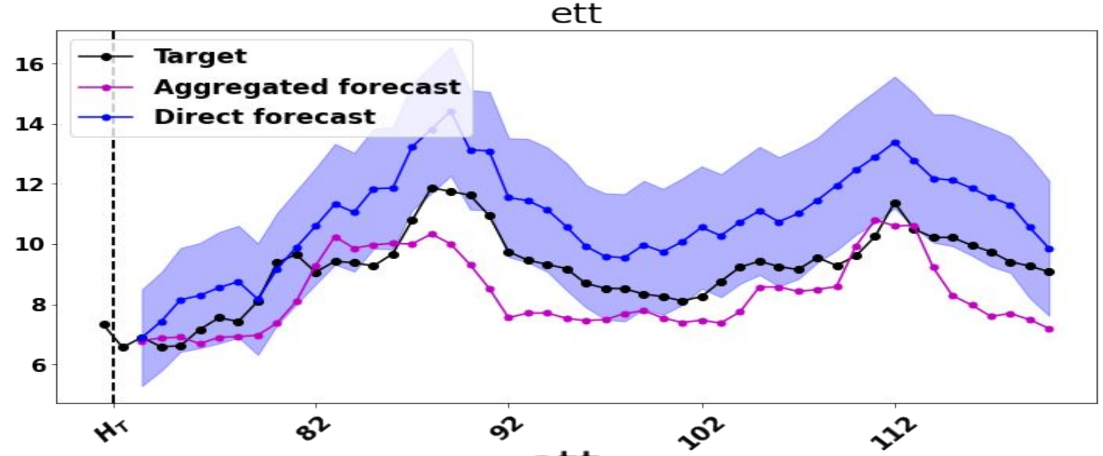
- Long-range forecasting is more challenging
    - Computational limitations
    - Modelling dependencies over long range in both *history* and *forecast-horizon*.

Aggregates of Forecasts

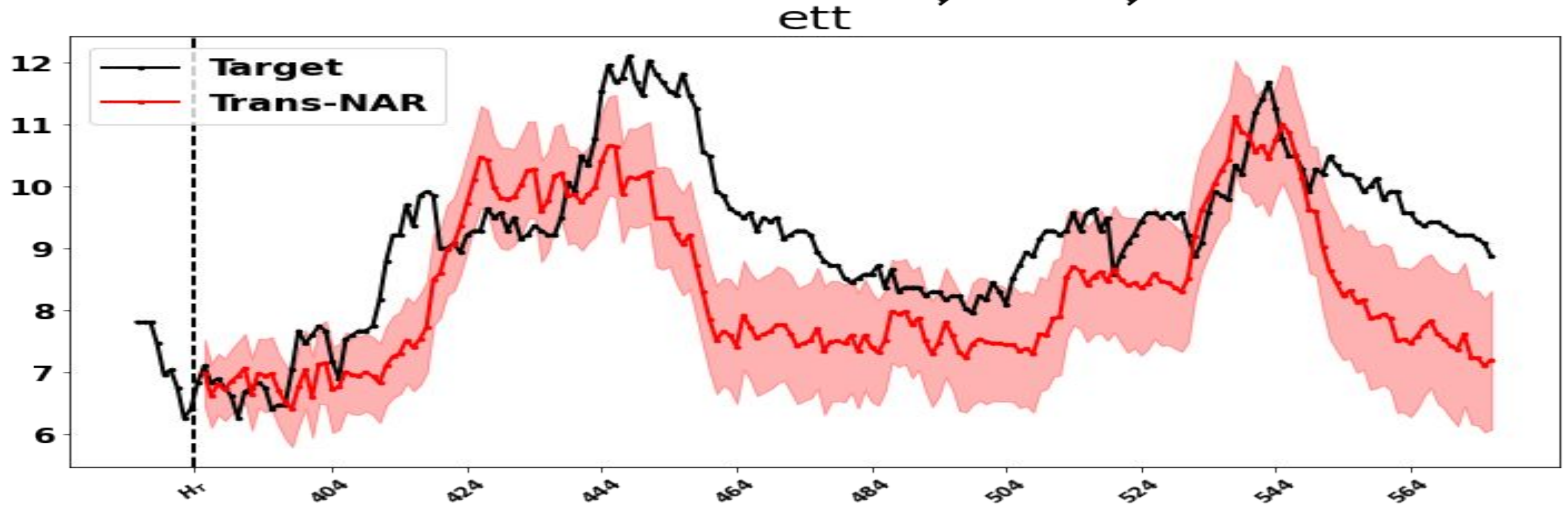
- Analysts are often interested in aggregated values of a window in a forecast horizon.
- For example,
  - Consider a demand forecasting task
  - Data contains daily sales
  - An analyst might want to look at monthly or quarterly forecasts for making a decision or creating a policy
- Depending on domain, other aggregations could be relevant, such as
  - Trend
  - Difference of sum
- Essentially, user/analyst could be interested in any aggregate depending on the domain and the specific business objective at the moment

Base-level Forecasts and Aggregate Forecasts





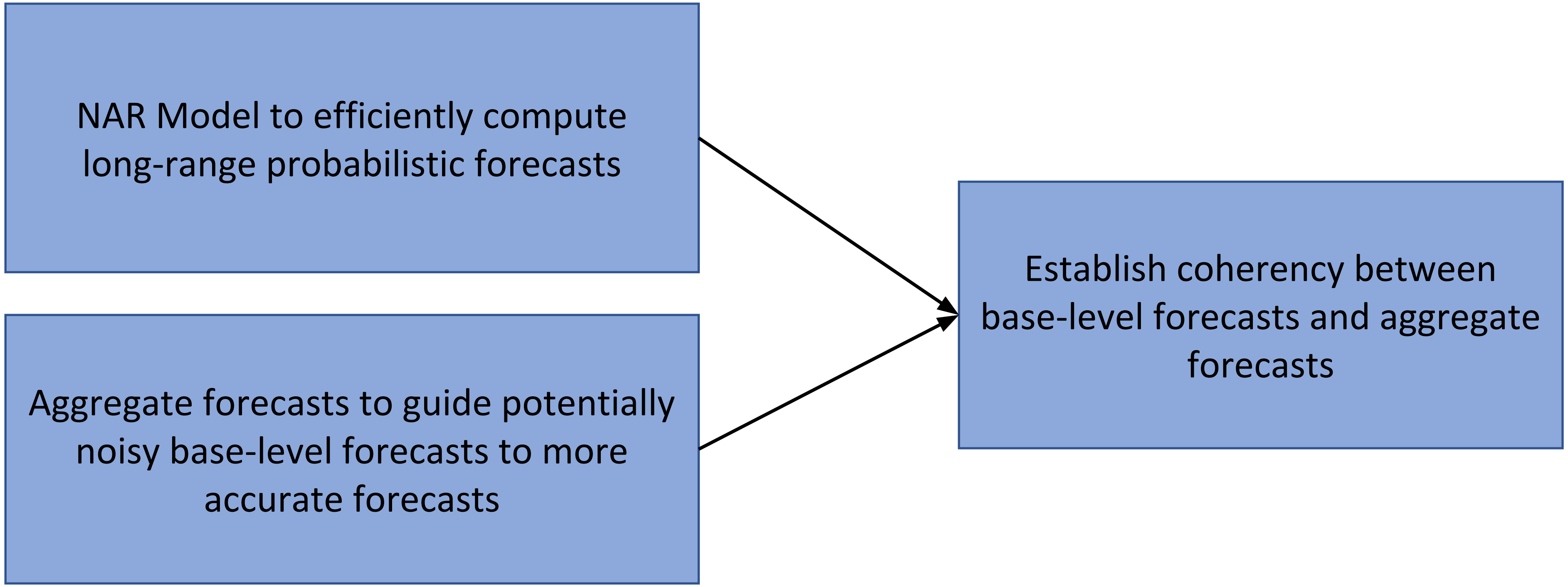
- We forecast a distribution at base level
- We can express aggregate also as a distribution obtained by aggregating base-level distributions.



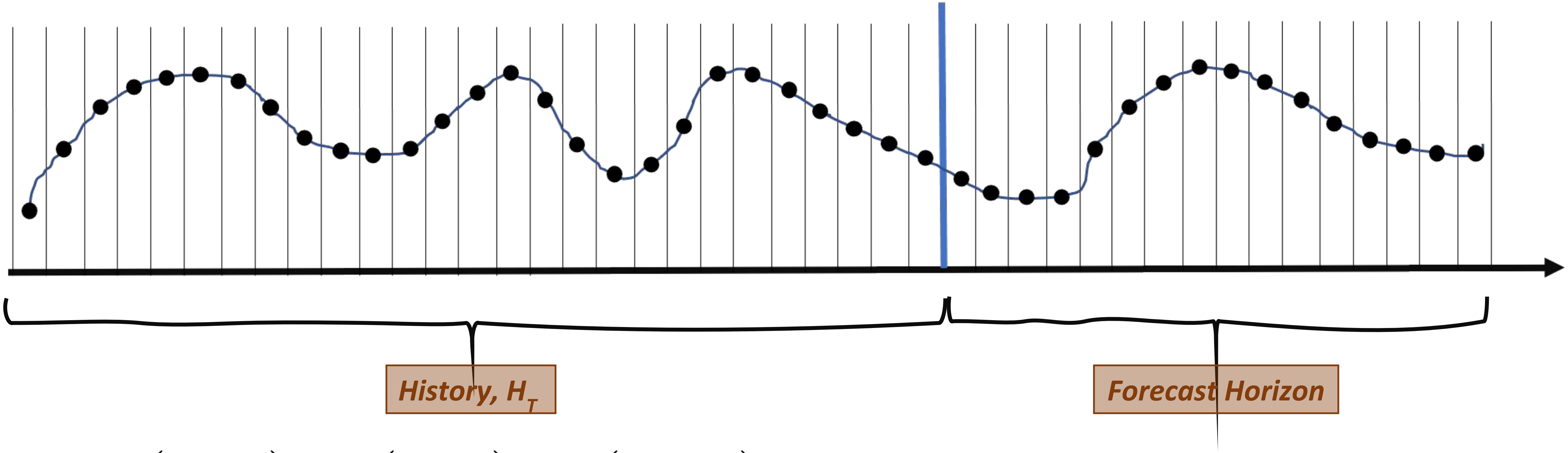
Non-autoregressive (NAR) Models for Long-Range Forecasting

- Auto-regressive models suffer from drift caused by cascading errors.
- Computing aggregate distributions using auto-regressive models require repeated sampling steps – Computationally Expensive
- NAR models offer an efficient way to calculate all values in the forecast-horizon
- NAR models have been shown to work well in practice.
- NAR models face a limitation when forecasting for a long-range:
    - Difficult to capture top-level patterns when time-series contains noise

Our Idea


NAR Model to efficiently compute long-range probabilistic forecasts
Aggregate forecasts to guide potentially noisy base-level forecasts to more accurate forecasts
Establish coherency between base-level forecasts and aggregate forecasts

Setup



$$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_t, y_t), \ldots, (\mathbf{x}_T, y_T)$$

$$(\mathbf{x}_{T+1}, y_{T+1}), \ldots, (\mathbf{x}_{T+R}, y_{T+R})$$

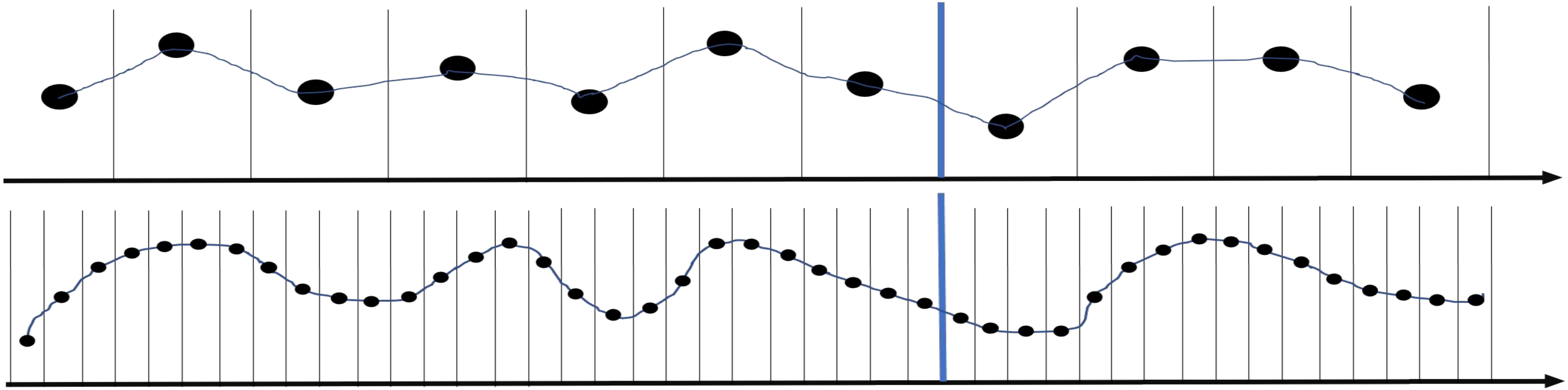
$$\mathbf{x}_t \in \mathbb{R}^d$$

denotes vector of input features

Series values at time *t*

Aggregate Functions

Average aggregate with window size (K) = 4



j-th value in i-th aggregated series, $$z_j^i = \mathbf{a}^i \cdot \mathbf{y}_{w_i,j} = \sum_{r=1}^{K_i} a_r^i \cdot y_{r+(j-1)K_i}$$

$\mathbf{a}^i \in \mathbb{R}^{K_i}$ denote vector of aggregation weights

Aggregate Functions

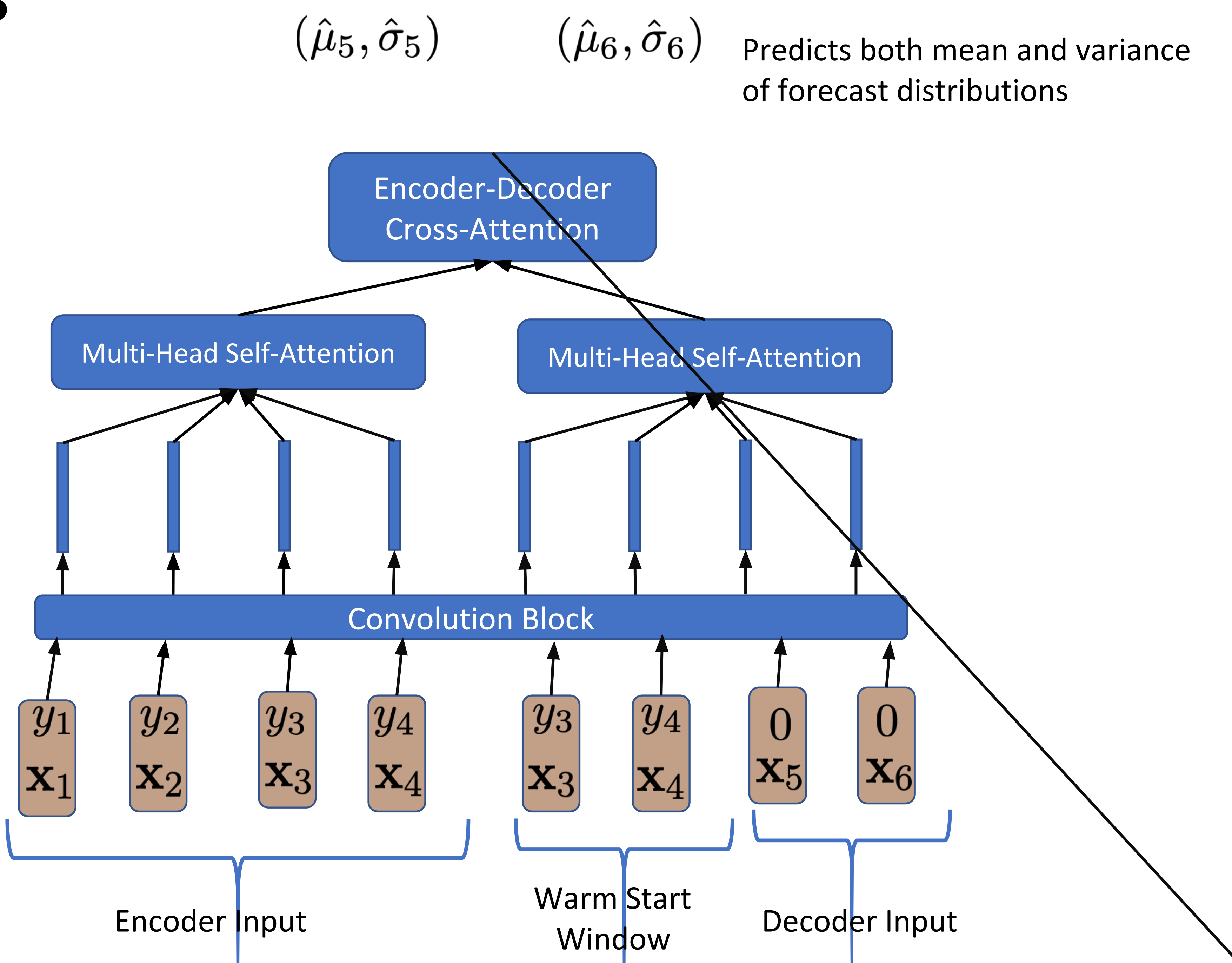
Average: $$z_j^i = \sum_{r=1}^{K_i} \underbrace{\frac{1}{K_i}}_{a_r^i} y_{(j-1)K_i+r}$$

Trend: $$z_j^i = \sum_{r=1}^{K_i} \underbrace{\left(\frac{r}{K_i} - \frac{K_i+1}{2K_i}\right)}_{a_r^i} \cdot y_{(j-1)K_i+r}$$

Our Forecasting Model Architecture

Predicts both mean and variance of forecast distributions

Convolution applied on a small window to extract representations that can be fed to the Transformer



Forecast Method

- For each aggregate (including original series), we train a separate forecast model.

Forecast distribution over *j-th* variable in *i-th* aggregated series:

$$\hat{P}(z_j^i | H_T, \mathbf{x}_j) \sim \mathcal{N}(\hat{\mu}(z_j^i), \hat{\sigma}(z_j^i))$$

- Since all aggregates are trained independently, the forecast distributions across aggregates are incoherent.

Coherent Forecasts

- In order to get the coherent forecasts, we infer a new consensus distribution *Q(.,.)* over base-level forecasts.

$$Q \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$$

where

$$\boldsymbol{\mu} = [\mu_{T+1}, \ldots, \mu_{T+R}]^T$$

Σ : Covariance Matrix of the joint distribution

- With this tractable form, we can compute the marginal distribution for aggregate variable z_j^i

$$Q_j^i = \mathcal{N}(\boldsymbol{\mu}_{w_i,j}^T \cdot \mathbf{a}^i, {\mathbf{a}^i}^T \cdot \Sigma_{w_i,j} \cdot \mathbf{a}^i)$$

Coherent Forecasts

- To establish coherence between marginals computed from *Q(..)* and forecast distributions $\hat{P}$*(..)*, we minimize the KL-distance as follows:

$$\min_{\boldsymbol{\mu},\Sigma} \sum_{i\in\mathcal{A}} \sum_{j=T_i}^{T_i+R_i} \alpha_i D_{\mathrm{KL}} \left(Q_j^i(z_j^i|\boldsymbol{\mu},\Sigma) \middle\| \hat{P}(z_j^i|\bullet) \right)$$

Values of $\boldsymbol{\mu}$ and Σ that minimize above objective are used as the final forecasts.

Solving the KL-distance Objective

$$\min_{\boldsymbol{\mu},\Sigma} \sum_{i\in\mathcal{A}} \sum_{j=T_i}^{T_i+R_i} \alpha_i D_{\mathrm{KL}}\left(Q_j^i(z_j^i|\boldsymbol{\mu},\Sigma) \middle|\middle| \hat{P}(z_j^i|\bullet)\right)$$

$$= D_{\mathrm{KL}}\left(\mathcal{N}\left(\boldsymbol{\mu}_{w_{i,j}}^T\mathbf{a}^i,\ \ {\mathbf{a}^i}^T\Sigma_{w_{i,j}}\mathbf{a}^i\right)\middle|\middle|\mathcal{N}\left(\hat{\mu}(z_j^i),\hat{\sigma}(z_j^i)\right)\right)$$

$$= \frac{\left(\boldsymbol{\mu}_{w_{i,j}}^T\mathbf{a}^i - \hat{\mu}(z_j^i)\right)^2 + \left({\mathbf{a}^i}^T\Sigma_{w_{i,j}}\mathbf{a}^i\right)}{2\hat{\sigma}(z_j^i)^2} - \log\frac{{\mathbf{a}^i}^T\Sigma_{w_{i,j}}\mathbf{a}^i}{\hat{\sigma}(z_j^i)^2}$$

Since both distributions are Gaussian, the KL-distance can be computed in closed form.

Rearranging the terms

$$= \frac{\left(\boldsymbol{\mu}_{w_{i,j}}^T\mathbf{a}^i - \hat{\mu}(z_j^i)\right)^2}{2\hat{\sigma}(z_j^i)^2} + \frac{\left({\mathbf{a}^i}^T\Sigma_{w_{i,j}}\mathbf{a}^i\right)}{2\hat{\sigma}(z_j^i)^2} - \log\frac{{\mathbf{a}^i}^T\Sigma_{w_{i,j}}\mathbf{a}^i}{\hat{\sigma}(z_j^i)^2}$$

*KL Distance between two Gaussians $D_{\mathrm{KL}}\left(\mathcal{N}(\mu_q,\sigma_q^2)\middle|\middle|\mathcal{N}(\mu_p,\sigma_p^2)\right) = \frac{(\mu_q-\mu_p)^2+\sigma_q^2}{2\sigma_p^2} - \log\frac{\sigma_q}{\sigma_p} - \frac{1}{2}$

Solving the KL-distance Objective

$$\min_{\boldsymbol{\mu},\Sigma} \sum_{i\in\mathcal{A}} \sum_{j=T_i}^{T_i+R_i} \alpha_i \frac{\left(\boldsymbol{\mu}_{w_{i,j}}^T \mathbf{a}^i - \hat{\mu}(z_j^i)\right)^2}{2\hat{\sigma}(z_j^i)^2} + \frac{\left(\mathbf{a}^{iT}\Sigma_{w_{i,j}}\mathbf{a}^i\right)}{2\hat{\sigma}(z_j^i)^2} - \log \frac{\mathbf{a}^{iT}\Sigma_{w_{i,j}}\mathbf{a}^i}{\hat{\sigma}(z_j^i)^2}$$

After expansion, optimization over mean and covariance form two independent optimization problems:

$$\boxed{\min_{\boldsymbol{\mu}} \sum_{i\in\mathcal{A}} \sum_{j=T_i}^{T_i+R_i} \frac{1}{\hat{\sigma}(z_j^i)^2}(\boldsymbol{\mu}_{w_{i,j}}^T \mathbf{a}^i - \hat{\mu}(z_j^i))^2}$$

$$\min_{\Sigma} \sum_{i\in\mathcal{A}} \sum_{j=T_i}^{T_i+R_i} \frac{\mathbf{a}^{iT}\Sigma_{w_{i,j}}\mathbf{a}^i}{2\hat{\sigma}(z_j^i)^2} - \log(\mathbf{a}^{iT}\Sigma_{w_{i,j}}\mathbf{a}^i)$$

Can be solved in closed form

Solving the KL-distance Objective (Solve for Covariance)

$$\min_{\Sigma} \sum_{i \in \mathcal{A}} \sum_{j=T_i}^{T_i+R_i} \frac{\mathbf{a}^{i^T} \Sigma_{w_{i,j}} \mathbf{a}^i}{2\hat{\sigma}(z_j^i)^2} - \log(\mathbf{a}^{i^T} \Sigma_{w_{i,j}} \mathbf{a}^i)$$

- Cannot be solved in closed form
- Number of parameters for Σ is R^2.
- In order to efficiently solve for Σ, we use low-rank approximation of Σ as follows:

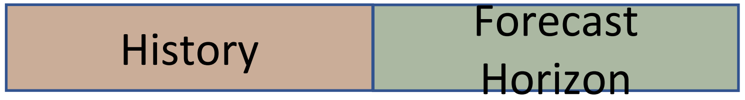
$$\hat{\boldsymbol{\Sigma}} = \begin{pmatrix} \sigma_{T+1}^2 & \cdots & 0 \\ & \ddots & \\ 0 & \cdots & \sigma_{T+R}^2 \end{pmatrix} + \begin{pmatrix} v_{T+1} \\ \vdots \\ v_{T+R} \end{pmatrix} \begin{pmatrix} v_{T+1} \\ \vdots \\ v_{T+R} \end{pmatrix}^T \quad \text{where} \quad v_{T+r} \in \mathbb{R}^k$$

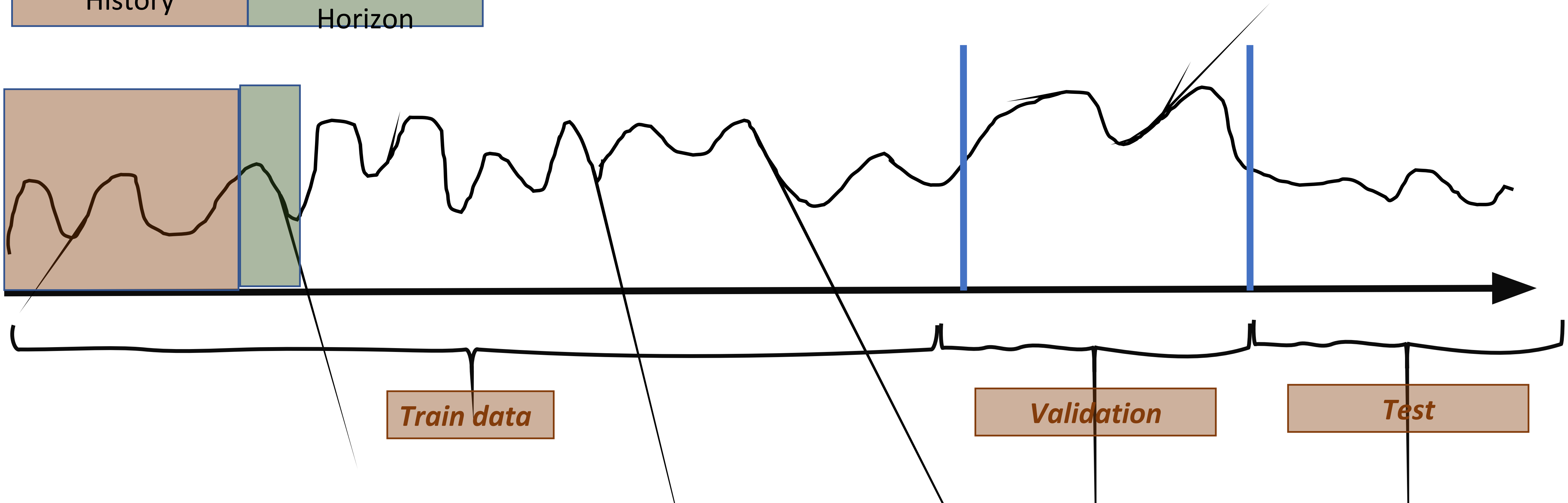
- Number of parameters using low-rank approximation is *O(R)*.
- Σ can be stored purely in the form of diagonal matrix and *V* vectors.

Training

- Large time-series is split into chunks of size *(T+R)*

A Chunk is denoted as follows:





Training Objective

$$\max_{\theta^i} \sum_{(\mathbf{x}_j^i, \mathbf{z}_j^i)} \sum_{t=T_i+1}^{T_i+R_i} \log \mathcal{N}(z_t; (\mu_t, \sigma_t) = F(H_T, \mathbf{x}, t|\theta^i))$$

θ^i : Parameters of *i-th* aggregate model

Datasets

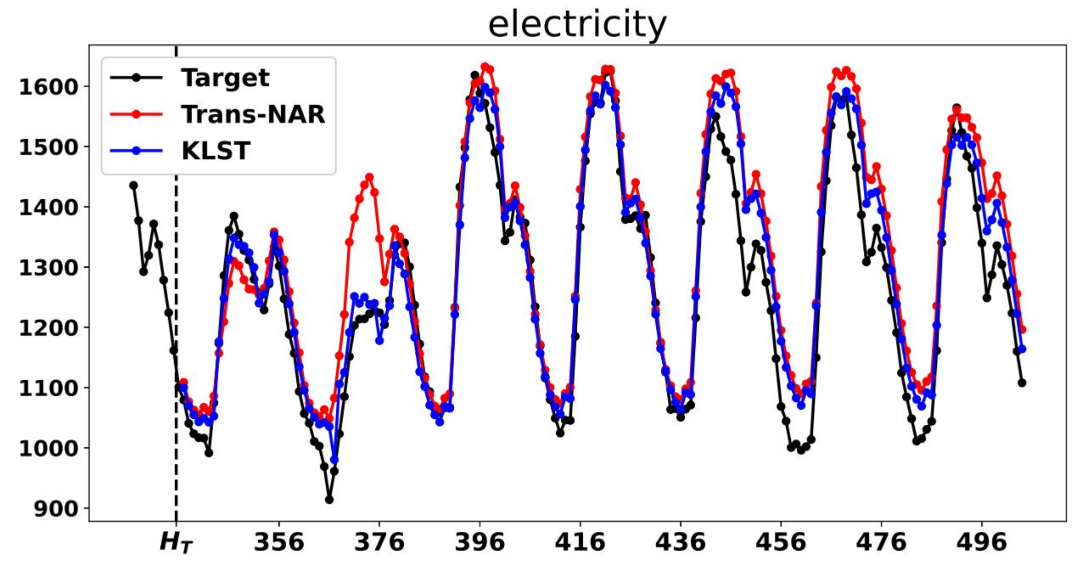
Dataset	# Series	Avg. T	R	train-len. /series	test-len. /series
ETT	1	384	192	55776	13824
ETTH	1	168	168	14040	3360
Electricity	1	336	168	36624	9072
Solar	137	336	168	7009	168

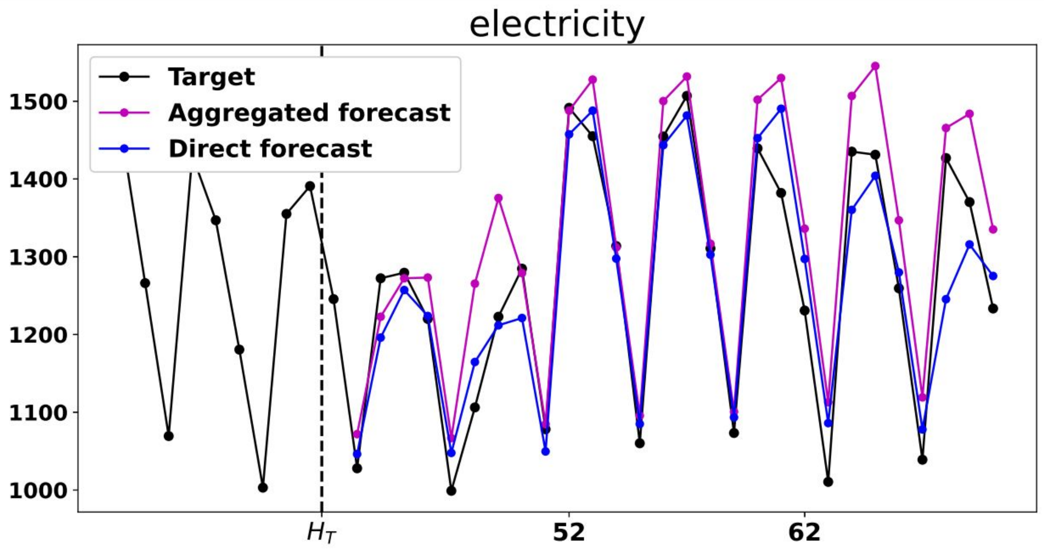
- ETTH, Electricity, and Solar are hourly datasets
- Whereas, ETT contains series collected over 15-minutes interval

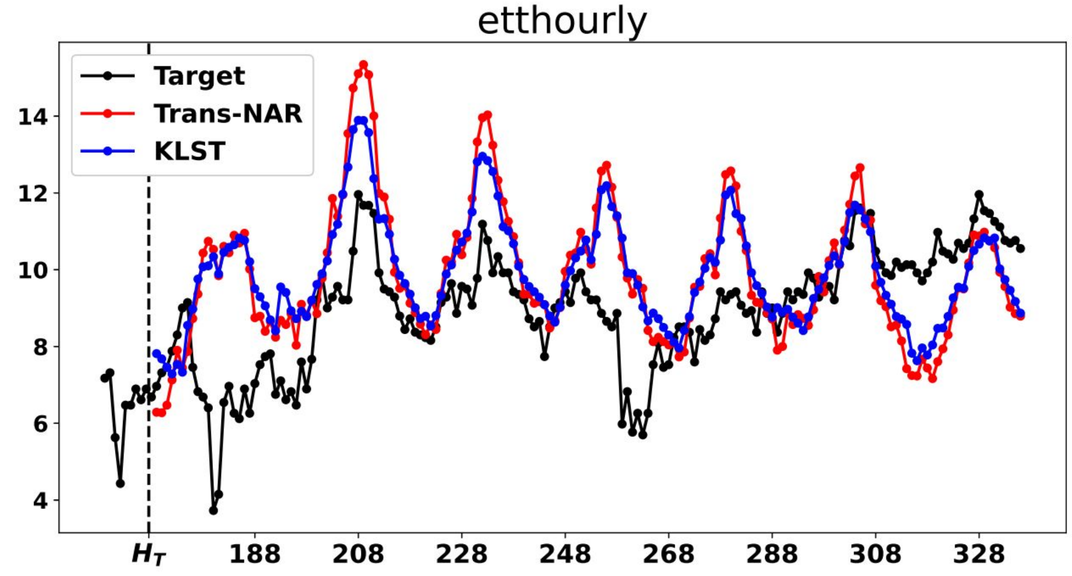
Methods Compared

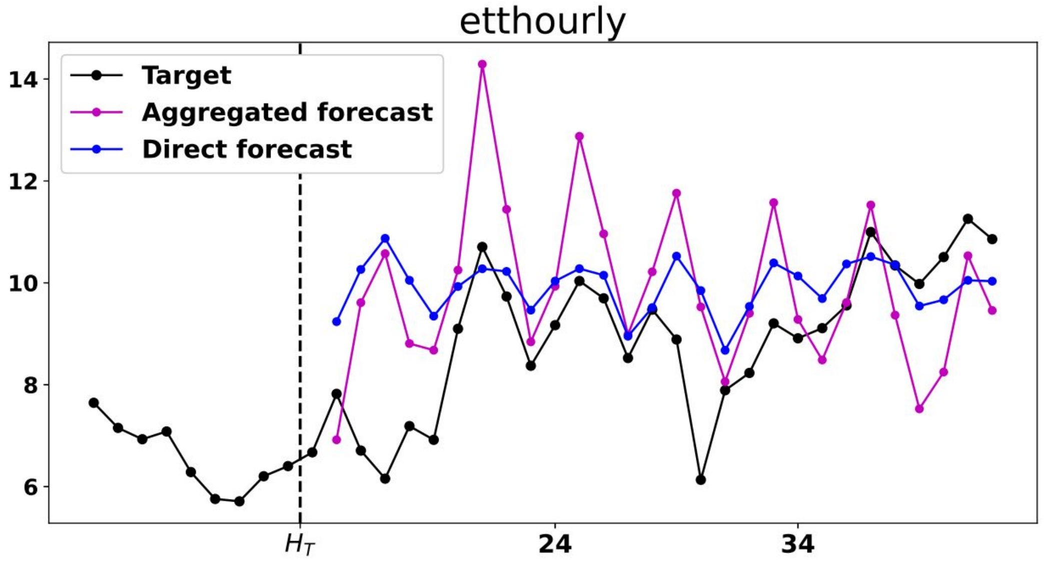
- **Informer** [1]: A transformer-based architecture that independently predicts values in the forecast horizon
- **Trans-NAR** : Our proposed architecture without KL-distance based inference
- **Trans-AR** : Auto-regressive version of our proposed architecture
- **KLST** : Trans-NAR + Our proposed inference method.

Anecdotes


electricity
Target
Trans-NAR
KLST
1600
1500
1400
1300
1200
1100
1000
900
H_T
356
376
396
416
436
456
476
496


electricity
Target
Aggregated forecast
Direct forecast
1500
1400
1300
1200
1100
1000
H_T
52
62


etthourly
Target
Trans-NAR
KLST
14
12
10
8
6
4
H_T
188
208
228
248
268
288
308
328


etthourly
Target
Aggregated forecast
Direct forecast
14
12
10
8
6
H_T
24
34

Evaluation Metrics

- Mean Absolute Error (MAE)
- Mean Squared Error (MSE)
- Continuous Ranked Probability Score

$$\Lambda_\alpha(q, y_t) = (\alpha - \mathcal{I}_{[y_t < q]})(y_t - q)$$

$$\mathrm{CRPS}(F_t^{-1}, y) = \int_0^1 2\Lambda_\alpha(F^{-1}(\alpha), y_t)d\alpha$$

Comparison with Baselines

Dataset Agg	Model	K: 1	4	8	12	24
ETT Sum	Informer	7.01	7.00	7.00	7.00	6.98
	Trans-AR	3.03	3.29	3.38	3.38	3.43
	Trans-NAR	1.25	1.36	1.39	1.39	1.38
	SHARQ	1.25	1.87	1.78	1.80	1.82
	KLST	**1.17**	**1.14**	**1.17**	**1.19**	**1.22**
ETT Slope	Trans-NAR	1.25	**0.13**	**0.07**	**0.06**	0.05
	KLST	**1.17**	0.30	0.12	**0.06**	**0.04**
ETT Diff	Trans-NAR	1.25	**0.14**	**0.16**	**0.20**	0.29
	KLST	**1.17**	0.33	0.26	0.25	**0.26**
Solar Sum	Informer	41.02	36.31	34.85	17.55	13.14
	Trans-AR	21.13	18.91	18.40	16.37	16.17
	Trans-NAR	13.85	13.25	12.95	12.78	12.43
	SHARQ	13.85	13.36	13.22	14.21	**11.60**
	KLST	**12.95**	**12.73**	**12.54**	**12.43**	12.21
Solar Slope	Trans-NAR	13.85	4.86	3.02	4.10	0.39
	KLST	**12.95**	**4.49**	**2.82**	**3.98**	**0.37**
Solar Diff	Trans-NAR	13.85	5.03	5.98	12.59	5.62
	KLST	**12.95**	**4.63**	**5.53**	**12.23**	**5.35**

Dataset Agg	Model	K: 1	4	8	12	24
ETTH Sum	Informer	4.80	4.77	4.73	4.67	4.57
	Trans-AR	1.96	2.01	1.98	2.01	1.96
	Trans-NAR	1.79	1.92	1.93	1.92	1.89
	SHARQ	1.79	1.91	1.73	1.75	1.78
	KLST	**1.64**	**1.61**	**1.65**	**1.67**	**1.69**
ETTH Slope	Trans-NAR	1.79	**0.26**	0.20	0.14	0.07
	KLST	**1.64**	0.37	**0.18**	**0.11**	**0.06**
ETTH Diff	Trans-NAR	1.79	**0.27**	0.39	0.46	0.50
	KLST	**1.64**	0.40	**0.37**	**0.39**	**0.41**
Elec Sum	Informer	172.3	159.7	155.8	118.1	109.6
	Trans-AR	140.2	137.8	134.0	109.6	104.7
	Trans-NAR	54.1	53.5	52.3	50.8	48.4
	SHARQ	54.1	**49.8**	**47.0**	50.5	46.3
	KLST	**50.2**	50.6	49.6	**48.4**	**46.2**
Elec Slope	Trans-NAR	54.1	8.96	6.25	5.65	2.23
	KLST	**50.2**	**8.26**	**5.76**	**5.18**	**2.14**
Elec Diff	Trans-NAR	54.1	9.50	13.23	18.37	16.13
	KLST	**50.2**	**8.80**	**12.22**	**16.84**	**15.44**

References

- Zhou, Haoyi, et al. "Informer: Beyond efficient transformer for long sequence time-series forecasting." *Proceedings of AAAI*. 2021. [1]