Characteristic Root Analysis and Regularization for Linear Time Series Forecasting

Zheng Wang, Kaixuan Zhang, Wanfang Chen, Xiaonan Lu,
Longyuan Li, Tobias Schlagenhauf

BOSCH

Theoretical Intuition
Linear Forecasting Model Generalization

We start our discussion with a simple noise-free timeseries

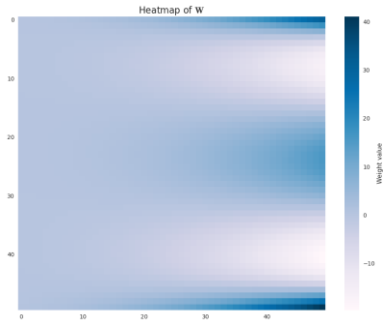
$$y_t^* = 0.01t^2 + \sin t$$

Linear Learnability:

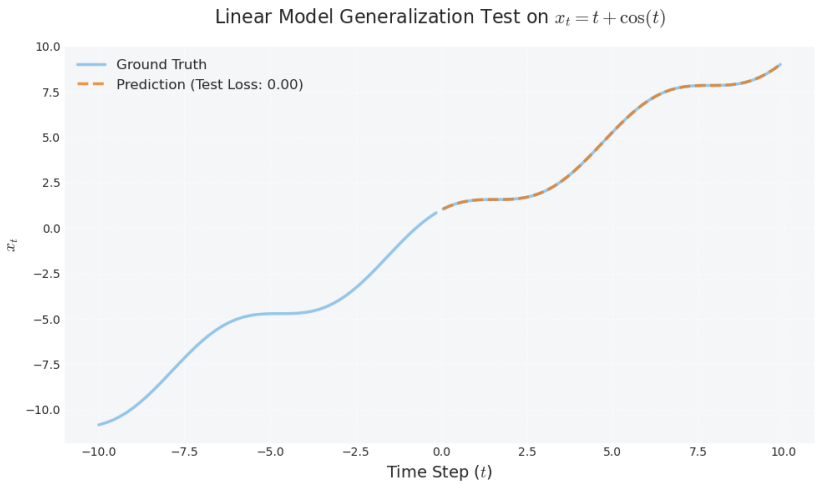
- We can ***perfectly*** fit a linear model with parameter $\mathbf{W}$ to forecast $y_t^*[\tau+1:\tau+L]$ with input $y_t^*[\tau-H+1:\tau]$.

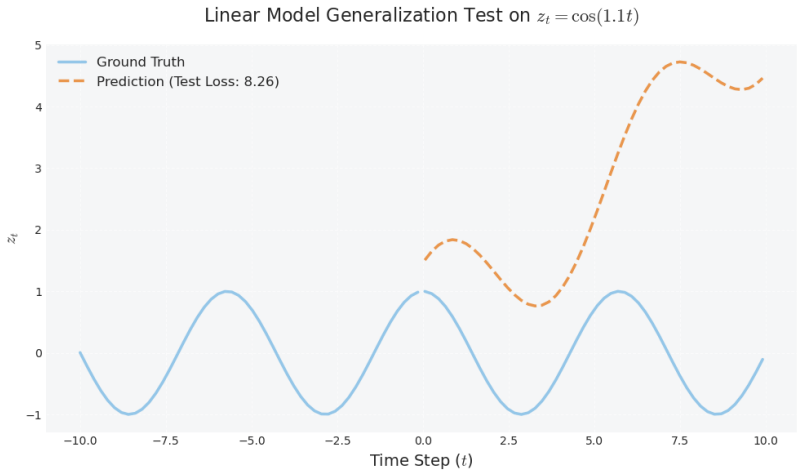
Generalization Behavior:

- $\mathbf{W}$ *can* also forecast $x_t^* = t + \cos t$.
- $\mathbf{W}$ *cannot* forecast $z_t^* = \cos 1.1t$.



Linear weight that fits y_t^*





2026-03-20

Theoretical Intuition
Noise-free Characteristic Roots Analysis

Backgrounds

- Linear Difference Equations:

$$y_t + a_1 y_{t-1} + \cdots + a_p y_{t-p} = 0$$

- Dynamic of linear difference equations are defined by *Characteristic Polynomials*

$$r^p + a_1 r^{p-1} + \cdots + a_p = 0$$

- Roots of the Characteristic Polynomial are called *Characteristic Roots*; they define the ***General Solutions*** of the system (for distinctive roots)

$$y_t = C_1 r_1^t + C_2 r_2^t + \cdots + C_p r_p^t$$

Recall the simple example

- For time series $y_t^* = 0.01t^2 + \sin t$, the roots are $e^i, e^{-i}, 1, 1, 1$.

- The general solution that representable by **W** is as follows, which represent all time series it can **forecast without any error**

$$y_t = C_1 e^{it} + C_2 e^{-it} + (C_3 + C_4 t + C_5 t^2) \cdot 1^t + \text{[general solution with other roots]}$$

Key Conclusions:

- Linear time series forecasting models can perform **zero-shot generalization** to any time series whose *characteristic roots* are a subset of its own.
- In noise-free condition, a sufficiently large (achieve training MSE of 0) linear model **always capture the real dynamics**.

Theoretical Intuition
Performance Upper Bound with Noise

In real cases, we almost always encounter noises, for instance:

$$\underbrace{y_t^* = 0.01t^2 + \sin t}_{\text{noise-free signal}} \quad \Rightarrow \quad \underbrace{y_t = y_t^* + \varepsilon_t}_{\text{observation with noise}}$$

Noise-based Error & Signal-based Error:

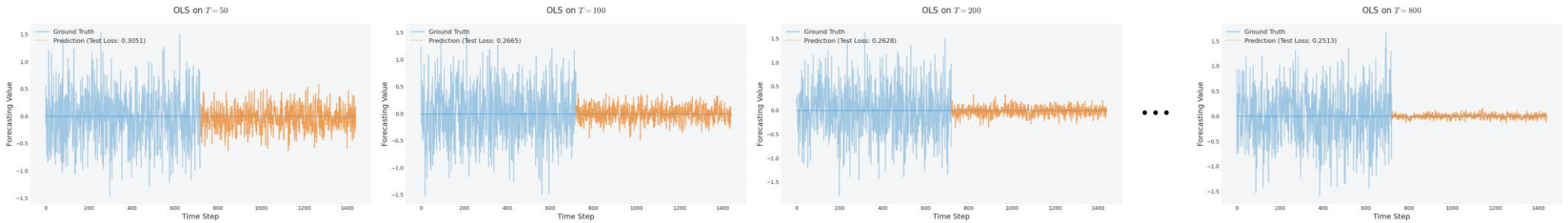
- The MSE we for a time series segment computed can be naturally decomposed into two parts

$$\mathbb{E}[\|(\mathbf{y}_{\text{fut}}^* - \mathbf{W}^\top \mathbf{y}_{\text{his}}^*) + (\boldsymbol{\varepsilon}_{\text{fut}} - \mathbf{W}^\top \boldsymbol{\varepsilon}_{\text{his}})\|_2^2]$$

- *Assumption:* we cannot forecast future noises and the best we can do is perfectly forecast the noise-free signal:

$$\mathbb{E}[\|(\mathbf{y}_{\text{fut}}^* - \mathbf{W}^\top \mathbf{y}_{\text{his}}^*) + (\boldsymbol{\varepsilon}_{\text{fut}} - \mathbf{W}^\top \boldsymbol{\varepsilon}_{\text{his}})\|_2^2] \geq \mathbb{E}[\|\boldsymbol{\varepsilon}_{\text{fut}} - \mathbf{W}^\top \boldsymbol{\varepsilon}_{\text{his}}\|_2^2] \quad \forall \mathbf{W}$$

- We focus on the lower bound $\mathbb{E}[\|\boldsymbol{\varepsilon}_{\text{fut}} - \mathbf{W}^\top \boldsymbol{\varepsilon}_{\text{his}}\|_2^2]$ evolves with dataset sizes:



Key Conclusions:

- Optimal MSE we can achieve is $\mathbb{E}[\|\boldsymbol{\varepsilon}_{\text{fut}}\|_2^2]$ when $\mathbf{W}^\top \boldsymbol{\varepsilon}_{\text{his}} = \mathbf{0}$ (noise with zero mean).
- When forecasting on noises with zero mean finite second moments, the $\mathbf{W}_{\text{OLS}}$ converges to population optimal $\mathbf{W}_\star$ with rate $\mathcal{O}(1/\sqrt{T})$.

2026-03-20

Methodologies
Suppressing Noise for Better Forecasting

Rank Reduction Methods

Even if clean signals has very few characteristic roots, data matrix typically becomes full rank due to noises. This motives *rank reduction* as a post-training denoising strategy.

Reduced Rank Regression (RRR):

- Find low-rank matrix $\mathbf{W}_\rho$ so that $\left\|\mathbf{W}_\rho^\top \mathbf{y}_{\text{his}} - \mathbf{W}^\top \mathbf{y}_{\text{his}}\right\|_F$ is smallest

Direct Weight Rank Reduction (DWRR):

- Find low-rank $\mathbf{W}_\rho$ so that $\left\|\mathbf{W}_\rho - \mathbf{W}\right\|_F$ is smallest

Algorithm 1 Reduced-Rank Regression

Input: $\mathbf{Y}_{\text{his}}, \mathbf{Y}_{\text{fut}}, \rho$;
Output: $\mathbf{W}$;
1: compute the OLS solution: $\mathbf{W}_{\text{OLS}} = (\mathbf{Y}_{\text{his}}^\top \mathbf{Y}_{\text{his}})^{-1} \mathbf{Y}_{\text{his}}^\top \mathbf{Y}_{\text{fut}}$;
2: calculate estimation : $\hat{\mathbf{Y}}_{\text{fut}} = \mathbf{Y}_{\text{his}} \mathbf{W}_{\text{OLS}}$
3: perform SVD on $\hat{\mathbf{Y}}_{\text{fut}}$: $\hat{\mathbf{Y}}_{\text{fut}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$;
4: truncate to rank ρ: $\mathbf{W}_{\text{RRR}} = \mathbf{W}_{\text{OLS}} \mathbf{V}_\rho \mathbf{V}_\rho^\top$;
5: **return** $\mathbf{W}_{\text{RRR}}$;

Algorithm 2 Direct Weight Rank Reduction

Input: $\mathbf{Y}_{\text{his}}, \mathbf{Y}_{\text{fut}}, \rho$;
Output: $\mathbf{W}$;
1: compute the OLS solution: $\mathbf{W}_{\text{OLS}} = (\mathbf{Y}_{\text{his}}^\top \mathbf{Y}_{\text{his}})^{-1} \mathbf{Y}_{\text{his}}^\top \mathbf{Y}_{\text{fut}}$;
2: perform SVD on $\mathbf{W}_{\text{OLS}}$: $\mathbf{W}_{\text{OLS}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$;
3: truncate to rank ρ: $\mathbf{W}_{\text{DWRR}} = \mathbf{U}_\rho \mathbf{\Sigma}_\rho \mathbf{V}_\rho^\top$;
4: **return** $\mathbf{W}_{\text{DWRR}}$;

Dynamic Rank Tuning – Root Purge

$\mathcal{G}_{\mathbf{W}}$: linear function with learnable parameter $\mathbf{W}$

$\mathcal{P}$: dimension aligner, e.g. padding or slicing

$$\min_{\mathbf{W}} \underbrace{\left\|\mathbf{Y}_{\text{fut}} - \mathcal{G}_{\mathbf{W}}(\mathbf{Y}_{\text{his}})\right\|_F^2}_{\text{root-seeking}} + \lambda \underbrace{\left\|\mathcal{G}_{\mathbf{W}} \circ \mathcal{P}\left(\mathbf{Y}_{\text{fut}} - \mathcal{G}_{\mathbf{W}}(\mathbf{Y}_{\text{his}})\right)\right\|_F^2}_{\text{root-purging}}$$

Rank-Nullity Theorem:

- $\text{Rank}(\mathbf{W}^\top) + \text{Nullity}(\mathbf{W}^\top) = \dim(\text{Domain}(\mathbf{W}^\top))$
- In human language: If a larger subspace of $\mathbf{W}^\top$'s domain is its null space ($\mathbf{x} \in \text{Null}(\mathbf{W}^\top)$ satisfy $\mathbf{W}^\top \mathbf{x} = \mathbf{0}$), then $\mathbf{W}$ is of low-rank.

Promoting a Close-to-Null Subspace:

- Recall that we cannot predict noise; thus, ideally, we wish to have $\mathbf{W}^\top \boldsymbol{\varepsilon}_{\text{his}} = \mathbf{0}$.
- We can estimate by $\boldsymbol{\varepsilon}_{\text{his}} = \mathbf{y}_{\text{fut}} - \hat{\mathbf{y}}_{\text{fut}}$. Since we cannot predict noise, $\hat{\mathbf{y}}_{\text{fut}}$ should be noise-free at convergence.
- We thus promote $\text{span}(\{\boldsymbol{\varepsilon}_{\text{his}}\})$ to behave **close to a null space**.

2026-03-20

Experiments
Main Results

Overall Results: Both methods we proposed can improve forecasting result comparing to previous methods.

Dataset	H	FEDformer	FilterNet	TSLANet	TimesNet	PatchTST	DLinear	SparseTSF	FITS	**RRR**	**Root Purge**
ETTh1	96	0.375	0.386	0.387	0.384	0.385	0.384	**0.362**	0.379	0.367	**0.359**
	192	0.427	0.420	0.421	0.436	0.413	0.443	0.403	0.414	**0.401**	**0.394**
	336	0.459	0.449	0.468	0.491	0.440	0.446	0.434	0.435	**0.430**	**0.423**
	720	0.484	0.500	0.529	0.521	0.456	0.504	0.426	0.431	**0.425**	**0.421**
ETTh2	96	0.340	0.309	0.299	0.340	0.274	0.282	0.294	0.272	**0.268**	**0.268**
	192	0.433	0.376	0.369	0.402	0.338	0.350	0.339	0.331	**0.329**	**0.328**
	336	0.508	0.418	0.390	0.452	0.367	0.414	0.359	**0.354**	**0.352**	0.355
	720	0.480	0.484	0.444	0.462	0.391	0.588	0.383	0.379	**0.376**	**0.377**
ETTm1	96	0.362	0.315	0.307	0.338	**0.292**	**0.301**	0.314	0.310	0.306	0.305
	192	0.393	0.364	0.349	0.374	**0.330**	0.335	0.343	0.338	0.336	**0.333**
	336	0.442	0.391	0.384	0.410	0.365	0.371	0.369	0.366	**0.365**	**0.360**
	720	0.483	0.457	0.471	0.478	0.419	0.426	0.418	0.415	**0.414**	**0.412**
ETTm2	96	0.189	0.180	0.197	0.187	0.163	0.171	0.165	0.163	**0.161**	**0.161**
	192	0.256	0.236	0.251	0.249	0.219	0.237	0.218	0.217	**0.216**	**0.216**
	336	0.326	0.292	0.303	0.321	0.276	0.294	0.272	0.269	**0.268**	**0.269**
	720	0.437	0.366	0.378	0.408	0.368	0.426	0.350	0.350	**0.348**	**0.350**
Weather	96	0.246	0.149	0.171	0.172	0.151	0.174	0.172	0.144	**0.140**	**0.142**
	192	0.292	0.194	0.219	0.219	0.195	0.217	0.215	0.188	**0.182**	**0.186**
	336	0.378	0.244	0.267	0.280	0.249	0.262	0.260	0.239	**0.232**	**0.238**
	720	0.447	0.316	0.332	0.365	0.321	0.332	0.318	**0.309**	**0.304**	0.310
Exchange	96	0.148	0.110	0.130	0.107	0.110	0.088	0.090	0.086	**0.084**	**0.082**
	192	0.271	0.230	0.243	0.226	0.284	0.182	0.182	0.177	**0.174**	**0.172**
	336	0.460	0.384	0.484	0.367	0.448	0.330	0.330	0.331	**0.324**	**0.324**
	720	1.195	1.062	1.079	0.964	1.092	1.060	1.051	**0.936**	**0.915**	0.941
Number of 1st Places		0	0	0	0	2	0	0	0	9	13

Verification of Theoretical Predictions

Singular Value Shrinkage and Scaling Property

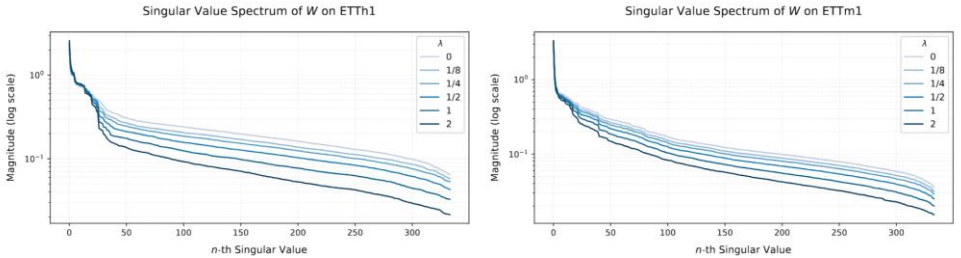


Figure 2: First 336 singular value magnitudes on ETTh1 and ETTm1 under different values of λ (log scale). As λ increases, Root Purge pushes the weight matrix $\mathbf{W}$ to have more smaller singular values, while the significant singular values remain largely unaffected.

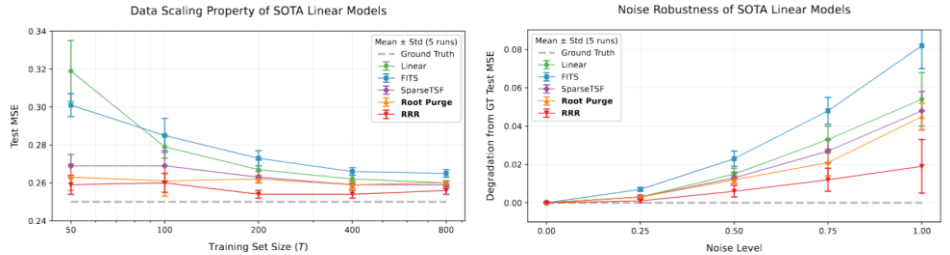


Figure 3: Data scaling and noise robustness of state-of-the-art linear time-series models. (left) RRR and Root Purge exhibit near-constant performance in data-scaling benchmarks. (right) Both methods exhibit robust performance under increasing noise levels, outperforming baseline models.

Singular Value Shrinkage:

In Figure 2, we can empirically confirm Root Purge should promote a subspace of $\mathrm{Domain}(\mathrm{W}^{\mathsf{T}})$ to behave closely to a null space.

Data Scaling & Noise Robustness

In Figure 3, we can empirically confirm Root Purge & RRR performs better at small training data and large noises.

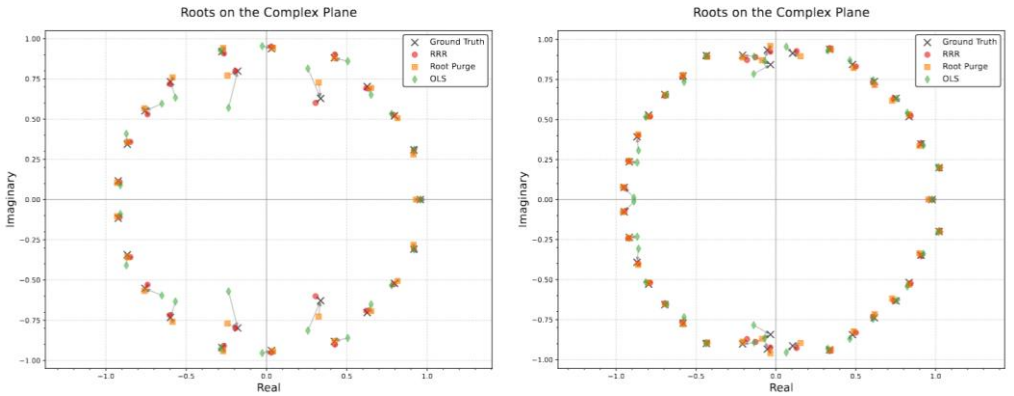
2026-03-20

BOSCH

Verification of Theoretical Predictions
Closer Characteristic Roots

Table 2: Comparison of Root Distance to Ground Truth across models (root pairings determined by Hungarian algorithm (Kuhn, 1955)). The results demonstrate the effectiveness of Rank Reduction and Root Purge in recovering roots closer to the true values compared to standard linear models.

Model	Root Distance to Ground Truth (mean ± std)
RRR	$\mathbf{0.036 \pm 0.014}$
Root Purge	$\mathbf{0.045 \pm 0.009}$
Standard Linear Model (OLS)	0.064 ± 0.025



(a) Root Distribution on the 1st forecasting horizon (b) Root Distribution on 12th forecasting horizon

Closer Characteristic Roots:

In Table 2, we can empirically confirm Root Purge & RRR produce characteristic roots that are closer to ground truth.

Root Visualizations:

In the left Figures, OLS characteristic roots also show visibly larger deviations from the ground truth.

Practical Robustness
Hyperparameter & Individual Channel Modeling (INC)

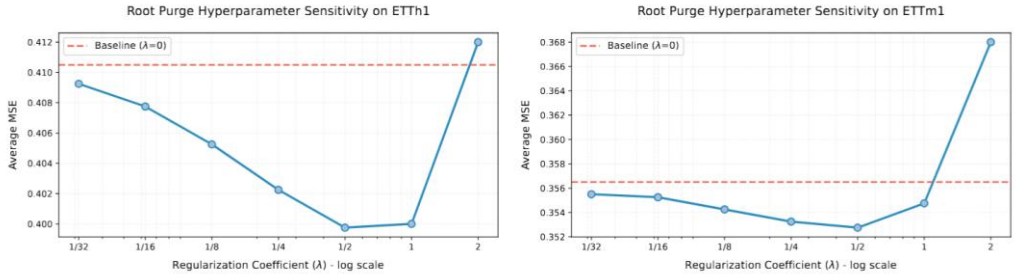


Figure 4: Average forecasting MSE on ETTh1 and ETTm1 across horizons $H = \{96, 192, 336, 720\}$ for different values of λ. Results indicate that a wide range of λ improves predictions, whereas larger values may cause over-regularization. A break-down table for each horizon is in Appendix E.7.

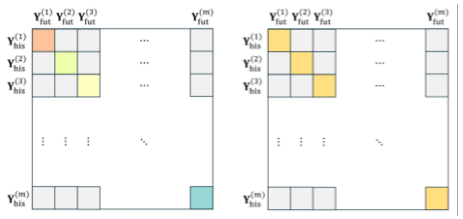
Hyperparameter Sensitivity:

In Figure 4, we can see that Root Purge improves performance across a wide range of λ values, making it easy to tune in practice.

Extending to Individual Channel Modeling (INC) :

While Channel Independence (CI) has be a common practice for time series modeling. We show that Root Purge can also work with INC setting, which fits separate model for each time series channel. We show only under root purge can we utilize this expressive power for better performances.

Dataset	ETTh1				ETTh2				ETTm1				ETTm2			
Horizon	96	192	336	720	96	192	336	720	96	192	336	720	96	192	336	720
(CI) Linear	0.375	0.411	0.439	0.431	0.270	0.331	0.354	0.378	0.306	0.336	0.365	0.414	0.162	0.217	0.269	0.350
(CI) Root Purge	0.359	0.394	**0.423**	**0.423**	**0.269**	0.329	0.355	0.377	0.305	0.333	0.360	**0.413**	0.162	0.217	**0.269**	**0.350**
INC Linear	0.397	0.432	0.450	0.453	0.290	0.338	0.364	0.383	0.297	0.337	0.370	0.421	0.165	0.222	0.276	0.356
INC Root Purge	**0.357**	**0.394**	0.427	0.438	0.271	**0.322**	**0.353**	**0.376**	**0.292**	**0.329**	**0.360**	0.418	**0.161**	**0.217**	0.273	0.356



(a) Linear INC (b) Linear CI

2026-03-20

BOSCH