ROAR: RANDOM-ORDER AUTOREGRESSIVE TIME SERIES FORECASTING FROM A VISUAL PERSPECTIVE

Anonymous authors
Paper under double-blind review

ABSTRACT

Autoregressive (AR) models have been increasingly applied to time series forecasting, owing to their favorable scalability and ability to accommodate prediction tasks of different horizons. However, most existing AR time series models are constrained by inflexible AR modeling paradigms and struggle with error accumulation. More general and flexible AR models and training methods for time series remain underexplored. To bridge this gap, we propose **ROAR**, a novel AR framework for time series forecasting. For AR modeling, motivated by the additional spatial structure introduced by the visual representation of time series, we reformulate forecasting as a random-order AR generation task on the time series image. For training, we devise a coupled training strategy that jointly supervises the entire backbone with signals from both teacher-forcing and free-running. These compact designs not only enable ROAR to learn richer contextual dependencies, but also narrow the training–inference mismatch, thereby alleviating the error accumulation incurred by the exposure bias. Comprehensive experiments conducted on 10 real-world benchmarks demonstrate that ROAR consistently achieves state-of-the-art performance while exhibiting superior robustness and generalization across multiple domains. Our code is available at: `https://anonymous.4open.science/r/ROAR-6702`.

1 INTRODUCTION

Time series forecasting supports applications in weather, transportation, healthcare, finance, energy, and AIOps (Price et al., 2025; Jin et al., 2024a; Ji et al., 2025; Shi et al., 2026; Parvathareddy et al., 2025; Zhang et al., 2025). Recent deep forecasting models have improved accuracy by combining neural architectures with task-specific inductive biases (Wang et al., 2026; Ding et al., 2026; Qiu et al., 2025; Liu et al., 2024b). In parallel, the success of autoregressive (AR) modeling in language and vision has motivated decoder-only forecasters that predict future tokens sequentially (Xiong et al., 2024; Liang et al., 2024; Liu et al., 2024d; Ansari et al., 2024). Their sequential formulation scales to different context and prediction lengths, making AR modeling a natural fit for general-purpose forecasting.

Most AR time series models nevertheless inherit a single chronological generation order from the underlying one-dimensional sequence. This formulation restricts training to one token trajectory, whereas image AR models can exploit two-dimensional structure to organize generation in more varied ways (Tian et al., 2024; Kumbong et al., 2025). AR forecasting also faces a train–inference mismatch: teacher forcing conditions on ground-truth prefixes during training, but inference conditions on the model's own predictions (Williams & Zipser, 1989; Ranzato et al., 2015). Errors can therefore propagate through recursive roll-outs, especially when long horizons require many generation steps.

Endogenous visual representations provide a way to revisit both issues. By folding a periodic time series into a two-dimensional image, they preserve the temporal signal while introducing a spatial grid of visual tokens (Chen et al., 2025). Existing vision-enhanced forecasters primarily use MAE-style reconstruction with a predefined prediction pattern (Chen et al., 2025; Shen et al., 2025), leaving open whether this spatial layout can support a more flexible AR factorization. Unlike a temporal token sequence with a prescribed order, this grid admits multiple serialization orders while

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

each token retains its physical position. This property suggests an AR forecaster that learns from diverse generation trajectories rather than a single chronological one.

Based on this observation, we propose **ROAR**, a **R**andom-**O**rder **A**uto**R**egressive framework for time series forecasting from a visual perspective. As illustrated in Figure 1, ROAR renders a time series as an endogenous image and, during training, autoregressively predicts its future visual tokens under randomly sampled serialization orders. The positional embedding remains tied to each token's physical location, so randomization changes the generation trajectory without discarding spatial identity. The same target is consequently learned under multiple causal factorizations of the visual tokens. The generated tokens are then restored to their physical order and decoded into the forecast. ROAR is trained with two complementary branches. The teacher-forcing branch predicts tokens from ground-truth prefixes in parallel, while the free-running branch conditions recursively on the model's own outputs and mirrors inference. Both branches supervise the same predictor in the series space; their joint optimization exposes it to inference-time inputs during training and reduces error propagation across AR steps. Across eight real-world benchmarks, ROAR achieves state-of-the-art performance, outperforming both vision-based and other strong baselines.

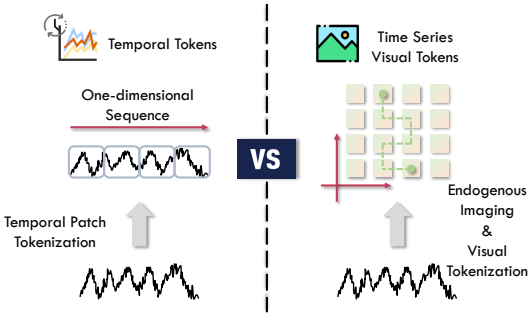

Figure 1: *Temporal Tokens* vs. *Time Series Visual Tokens*. The two-dimensional layout of visual tokens supports random-order AR generation while preserving each token's physical position.

Our contributions are summarized as follows:

- We formulate time series forecasting as random-order AR generation over endogenous visual tokens, using their two-dimensional layout to diversify generation trajectories.
- We introduce a dual-branch objective that combines teacher forcing with free-running supervision for random-order AR training.
- We evaluate ROAR on 10 benchmarks across multiple domains and show strong average accuracy, particularly for horizons requiring repeated AR roll-outs.

2 Background and Preliminaries

2.1 Autoregressive Time Series Forecasting

Given historical observations $\mathcal{X} \in \mathbb{R}^{C\times O}$, time series forecasting predicts a future sequence $\mathcal{Y} \in \mathbb{R}^{C\times F}$, where C, O, and F denote the numbers of channels, historical steps, and future steps, respectively. ROAR follows the channel-independent setting (Nie et al., 2023); we therefore omit C below and describe one channel at a time. An AR model factorizes the joint distribution of a token sequence $\mathbf{X} = \{x_1, \ldots, x_L\}$ according to a chosen order:

$$p(\mathbf{X}) = \prod_{i=1}^{L} p(x_i \mid x_{<i}). \tag{1}$$

DeepAR (Salinas et al., 2020) and SegRNN (Lin et al., 2026) established recurrent and segment-wise AR forecasting, while recent decoder-only models, including Timer (Liu et al., 2024d; 2025), Chronos (Ansari et al., 2024), TimesFM (Das et al., 2024), and Time-MoE (Shi et al., 2025), scale next-token prediction to larger datasets and models. These methods serialize tokens in temporal order. ROAR instead studies AR factorization over spatially organized visual tokens while retaining the physical position of every token.

2.2 Vision-Enhanced Time Series Forecasting

Vision-enhanced forecasters construct endogenous images from time series and apply visual representation learning to forecasting (Jiang et al., 2025). VisionTS (Chen et al., 2025) casts forecasting

as image reconstruction with a pre-trained masked autoencoder (He et al., 2022); DMMV (Shen et al., 2025) jointly models temporal signals and endogenous images; LDM4TS (Ruan et al., 2025) uses multi-view visual representations; and TimeVLM (Zhong et al., 2025) incorporates a vision-language backbone. Following VisionTS, ROAR standardizes a single-channel history $x \in \mathbb{R}^O$, folds it by its dominant period P, and resizes the resulting matrix into a history image I_h. A blank future region I_f is appended on the right, yielding an endogenous image $I = [I_h, I_f]$ whose geometry is detailed in Appendix B.2. Existing methods mainly recover the future region through masked reconstruction under a predefined prediction pattern. ROAR uses the same type of endogenous representation but treats its two-dimensional layout as a set of positioned tokens over which multiple AR orders can be defined.

2.3 Problem Formulation: Random-Order Autoregression

Let $\mathbf{Z}_h = \{z_1^h, \ldots, z_N^h\}$ and $\mathbf{Z}_f = \{z_1^f, \ldots, z_M^f\}$ denote the visual tokens of the history and future regions. For permutations π_h and π_f of their token indices, ROAR models the conditional distribution as

$$p_\theta(\mathbf{Z}_f \mid \mathbf{Z}_h, \pi_h, \pi_f) = \prod_{i=1}^{M} p_\theta\left(z^f_{\pi_f(i)} \mid \mathbf{Z}_h^{\pi_h}, \mathbf{Z}_{f,\pi_f(<i)}\right), \tag{2}$$

where $\mathbf{Z}_h^{\pi_h}$ is the permuted history sequence and $\mathbf{Z}_{f,\pi_f(<i)}$ contains the future tokens preceding step i in the sampled order. Each positional embedding remains attached to its token's physical location, so a permutation changes which token is predicted next without changing where that token belongs in the image. Training over sampled permutations exposes the predictor to multiple causal factorizations of the same target. Choosing a deterministic future permutation recovers fixed-order autoregression, whereas sampling permutations trains the predictor across multiple factorizations. Figure 8 provides a visual comparison with MAE-style reconstruction.

3 Methodology

Figure 2 presents ROAR, which instantiates the random-order formulation in Equation (2) in four stages. ① The input series is converted into an endogenous image whose left and right regions represent the observed history and the future to be predicted, respectively (Section 2.2). ② Frozen visual encoders map the history and future regions to historical prefix tokens and future target tokens during training (Section 3.1). ③ The two token sets are permuted and serialized for a causal Transformer, which predicts future tokens in the sampled order and is optimized by the dual-branch objective (Sections 3.2 and 3.3). ④ The generated tokens are restored to their physical positions, decoded into the future image region, and mapped back to the forecast sequence. ROAR also applies reversible normalization (Kim et al., 2022) before image construction and inverts it after decoding.

3.1 Visual Tokenization

Given the endogenous image $I = [I_h, I_f]$ introduced in Section 2.2, ROAR tokenizes its history and future regions with two frozen visual encoders. This produces the historical token set $\mathbf{Z}_h$ and future token set $\mathbf{Z}_f$ used by Equation (2). The image construction, normalization, and exact token geometry are provided in Appendix B.2.

Historical Visual Token Encoder. Following the visual tokenization used by VisionTS-style forecasters (Chen et al., 2025), the rendered image is partitioned into $h \times w$ patches and encoded into a grid of latent tokens. A frozen history encoder $E_h(\cdot)$ maps the history region I_h to N historical visual tokens $\mathbf{Z}_h = \{z_1^h, z_2^h, \ldots, z_N^h\}$, which serve as the prefix context for AR generation.

Future Visual Token Encoder. A frozen future encoder $E_f(\cdot)$ maps the future region I_f to M future visual tokens. During training, the future region rendered from the ground-truth observations provides the targets $\mathbf{Z}_f = \{z_1^f, z_2^f, \ldots, z_M^f\}$. The paired frozen decoder $D_f(\cdot)$ maps generated future tokens back to the pixel space. After generation, ROAR restores the tokens to their physical order, decodes the future image region, joins it with the observed history region, and inverts the resizing and normalization to recover the forecast sequence. Both visual encoders and the decoder are pre-trained by reconstruction and remain frozen during forecasting training.

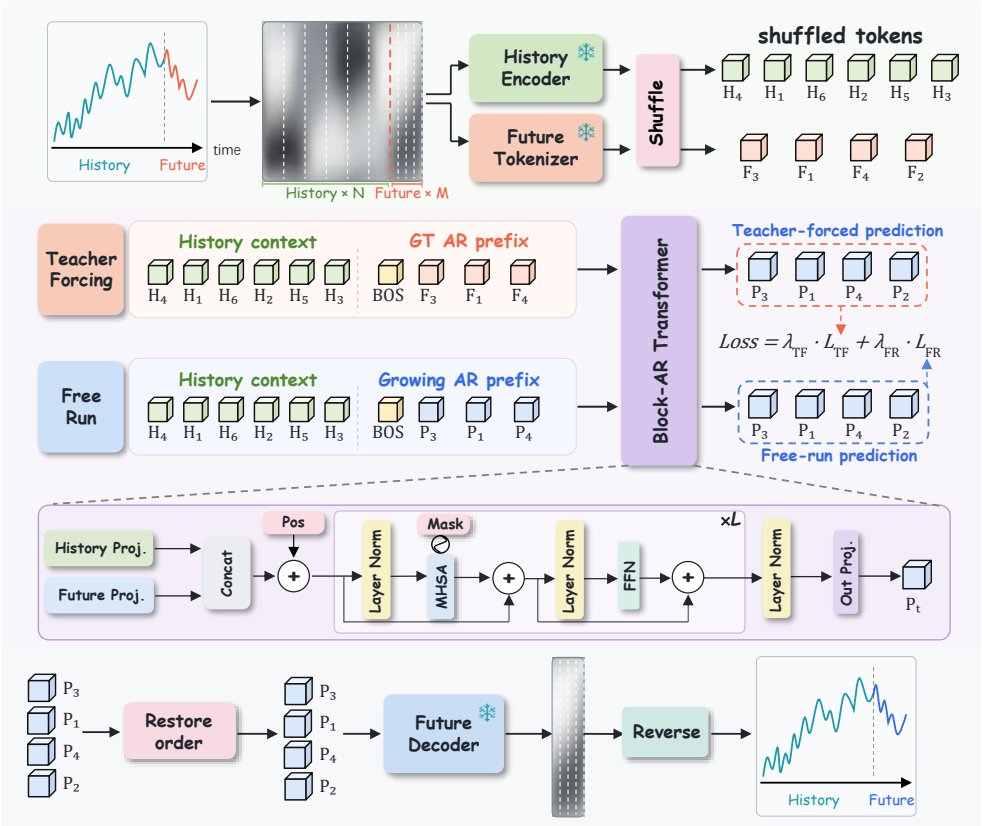

Figure 2: Overview of ROAR. The input series is rendered as an endogenous image, whose history and future regions are encoded into visual tokens by frozen encoders. Both token sets are permuted before entering the random-order predictor, which is jointly optimized with teacher-forcing and free-running branches. The generated future tokens are restored to their physical positions, decoded into the future image region, and mapped back to the forecast sequence.

3.2 Random-Order Token Predictor

ROAR serializes the historical and future visual tokens for a random-order causal predictor that implements Equation (2).

Random Token Ordering. For each training sample, ROAR independently draws a history permutation π_h and a future permutation π_f and serializes the corresponding tokens in those orders. Each positional embedding is bound to the token's physical location and is permuted together with that token. The permutation therefore changes the serialization order—which token is predicted next—without removing the token's spatial identity. A causal attention mask prevents each serialized position from attending to later positions in the sampled order. At inference, ROAR uses a fixed history permutation and generates future blocks from left to right, making the forecast deterministic for a fixed checkpoint.

Block-wise Causal Transformer. The serialized tokens are processed by a block-wise causal Transformer $G_\theta(\cdot)$. Separate linear projections first map the historical and future tokens into a shared latent space, and a learnable BOS token marks the start of future-token generation. The resulting sequence is passed through L pre-norm layers (Vaswani et al., 2017), each containing masked multi-head self-attention and a feed-forward network. Under the causal mask, the token at generation step i is predicted from the entire serialized history and the earlier future tokens:

$$\left(\mathbf{U}_h, \hat{z}^f_{\pi_f(i)}\right) = G_\theta\left(\mathbf{Z}_h^{\pi_h}, \mathrm{BOS}, \mathbf{Z}_{f,\pi_f(<i)}\right), \tag{3}$$

where $\hat{z}^f_{\pi_f(i)}$ is obtained by applying the output projection to the final latent position. The Transformer also produces historical-position latents $\mathbf{U}_h$, but they are not passed through the output projection and receive no direct prediction loss. Sampling a new pair (π_h, π_f) for every training example optimizes Equation 2 over multiple generation orders rather than a single serialization.

3.3 Dual-Branch Training

Teacher forcing conditions each prediction on ground-truth prefix tokens and supports parallel AR training (Williams & Zipser, 1989). During inference, the model instead conditions on its own preceding predictions; this difference is commonly described as exposure bias (Ranzato et al., 2015). ROAR trains the same random-order predictor with two branches that share the sampled permutations (π_h, π_f), as shown in Figure 3.

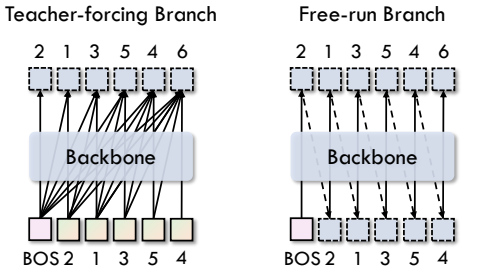

Figure 3: Dual-branch training. The teacher-forcing branch conditions on preceding ground-truth tokens, whereas the free-running branch conditions on tokens generated by the model. Both branches supervise the same random-order predictor and share the sampled permutations.

Teacher-Forcing Branch. The teacher-forcing branch supplies the ground-truth future tokens as prefix context, allowing all M predictions $\{\hat{z}^f_{\pi_f(i)}\}_{i=1}^M$ to be produced in one parallel forward pass.

Free-Running Branch. The free-running branch starts from the BOS token and recursively generates the future tokens conditioned on its own preceding predictions, matching the autoregressive dependency used at inference.

ROAR decodes the predictions of both branches into the series space through the inverse mapping in Section 3.1 and computes an L2 loss against the ground-truth future sequence $y \in \mathbb{R}^F$:

$$\mathcal{L} = \lambda_{\mathrm{TF}} \big\| \hat{y}_{\mathrm{TF}} - y \big\|_2^2 + \lambda_{\mathrm{FR}} \big\| \hat{y}_{\mathrm{FR}} - y \big\|_2^2, \tag{4}$$

where $\hat{y}_{\mathrm{TF}}$ and $\hat{y}_{\mathrm{FR}}$ are the decoded forecasts, and λ_{TF} and λ_{FR} balance the two objectives. Because the visual encoders and decoder are frozen, both losses update only the block-wise causal Transformer, while supervision remains aligned with the final forecasting target.

4 Experiments

4.1 Datasets

We evaluate ROAR on 10 real-world benchmark datasets spanning several domains. These benchmarks are ETT (ETTh1, ETTh2, ETTm1 and ETTm2) (Zhou et al., 2021), Weather (Wu et al., 2021), Exchange (Lai et al., 2018), Solar (Lai et al., 2018; Wu et al., 2021), Traffic (Wu et al., 2021), Environment (Liu et al., 2024a), and Wind (Li et al., 2022). More information about datasets is provided in Appendix A.1.

4.2 Baselines

We select nine recent state-of-the-art deep forecasting models as baselines to provide a comprehensive evaluation. These baselines include image-guided, text-guided, autoregressive, and purely numerical forecasting models, including VisionTS (Chen et al., 2025), Timer-XL (Liu et al., 2025), AutoHFormer (Zhang et al., 2026b), TimeLLM (Jin et al., 2024b), TimeMixer (Wang et al., 2024), TimeTK (Zhang et al., 2026a), iTransformer (Liu et al., 2024b), PatchTST (Nie et al., 2023), and TimesNet (Wu et al., 2023). Detailed information about these baselines is provided in Appendix A.2.

4.3 Setup

Our experimental settings follow widely recognized works (Liu et al., 2024d; 2025). We fix the input context length to 672 and set the future window to 96 (Liu et al., 2025; 2024c). For all benchmark datasets, we consider four prediction horizons: {96, 192, 336, 720}. When the required forecasting

horizon exceeds the future window, we perform autoregressive rolling generation during inference until the required prediction horizon is reached. More details are provided in Appendix A.3.

Table 1: The results of the full-shot experiments. The input context length is fixed at 672, and the prediction horizons are set to {96, 192, 336, 720}, respectively. A single model is trained for all prediction horizons. We report the average MSE and MAE of four prediction horizons. Best and second-best results are marked in **bold** and underlined, respectively.

Models	ROAR (**Ours**)		VisionTS (2025)		TimeLLM (2024b)		TimeMixer (2024)		TimeTK (2026a)		Timer-XL (2025)		AutoHFormer (2026b)		iTransformer (2024b)		PatchTST (2023)		TimesNet (2023)	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	**0.348**	0.385	0.359	**0.378**	0.366	0.398	0.391	0.408	0.377	0.398	0.365	0.393	0.350	0.386	0.384	0.409	0.358	0.393	0.399	0.406
ETTm2	0.276	0.333	0.282	0.334	0.281	0.331	0.289	0.338	0.312	0.364	0.273	**0.323**	**0.272**	0.329	0.285	0.340	0.273	0.331	0.314	0.333
Exchange	**0.328**	**0.392**	0.377	0.420	0.441	0.459	0.492	0.486	0.361	0.410	0.851	0.611	0.434	0.447	0.422	0.442	0.411	0.434	0.993	0.659
Weather	0.242	0.283	**0.220**	**0.254**	0.240	0.281	0.228	0.268	0.237	0.276	0.258	0.292	0.234	0.271	0.254	0.286	0.229	0.269	0.264	0.293
Solar	0.214	0.262	0.265	0.253	0.238	0.307	0.210	0.265	0.227	0.259	0.236	**0.238**	**0.204**	0.266	0.215	0.279	0.209	0.258	0.213	0.295
Traffic	0.421	0.291	0.406	0.261	0.428	0.329	0.393	0.272	0.506	0.366	0.407	**0.254**	0.423	0.309	**0.388**	0.272	0.391	0.275	0.602	0.322
Environment	**0.247**	**0.360**	0.271	0.388	0.268	0.372	0.254	0.370	0.255	0.367	0.284	0.392	0.258	0.361	0.305	0.414	0.268	0.384	0.258	0.366
Wind	**1.130**	**0.768**	1.151	0.770	1.164	0.786	1.387	0.840	1.153	0.771	1.260	0.805	1.173	0.780	1.502	0.879	1.140	0.775	1.305	0.809
Average	**0.401**	0.384	0.416	**0.382**	0.428	0.408	0.456	0.406	0.428	0.401	0.492	0.413	0.418	0.394	0.469	0.415	0.410	0.390	0.543	0.435

4.4 Main Results

Table 1 compares ROAR with 9 baselines on different benchmarks. All methods share the same protocol of a 672-step input, a 96-step output window, and autoregressive rolling for longer horizons, so the table isolates the modeling choice rather than the inference procedure. ROAR achieves the best average MSE (0.401) and the second-best average MAE (0.384), and ranks first on ETTm1 (MSE) and on Exchange, Environment, and Wind (MSE and MAE). Two comparisons are informative. **(1) Fixed versus random order over visual tokens.** VisionTS renders the same endogenous image as ROAR but fills the future region by MAE-style reconstruction along a single predefined pattern. ROAR attains lower MSE on six of the eight datasets and lowers the average MSE from 0.416 to 0.401. The largest gains appear on Exchange (0.377 $\rightarrow$ 0.328), Solar (0.265 $\rightarrow$ 0.214), and Environment (0.271 $\rightarrow$ 0.247). These series have weak or drifting periodicity, so a single predefined reconstruction pattern fits them poorly, and random-order generation over the future canvas helps most. **(2) Visual tokens versus temporal tokens.** The remaining baselines tokenize the raw series along the time axis and predict each 96-step block from those temporal tokens, whether the backbone is a Transformer (Timer-XL, AutoHFormer, PatchTST, iTransformer), an LLM (TimeLLM), or an MLP model (TimeMixer). ROAR obtains a lower average MSE than every temporal-token baseline. Against the two AR forecasters built for rolling generation, Timer-XL and AutoHFormer, ROAR lowers the average MSE by 18.5% and 4.1%, and it remains stable on Exchange, where Timer-XL degrades to 0.851. This comparison supports our central claim that the two-dimensional layout of visual tokens and randomized generation trajectories capture contextual dependencies that a single chronological order over temporal tokens cannot.

On Traffic, ROAR trails the best baseline by about 0.03 MSE; we discuss this gap in Appendix I.

Ultra-Long-Horizon Forecasting. Since ROAR is designed to alleviate error accumulation in recursive generation, its advantage should grow with the number of inference steps. Figure 4 compares ROAR with VisionTS, Timer-XL, and AutoHFormer at $H = 1080$ (12 recursive 96-step calls), and Table 7 in Appendix D.3 extends the comparison to $H = 2160$. ROAR attains the best average MSE and MAE at all four extended horizons, and its margin over VisionTS grows from 3.6% at the standard horizons to 4.9%–6.0%. For example, on ETTm2 the MSE of Timer-XL rises from 0.507 at $H = 1080$ to 0.643 at $H = 2160$, whereas ROAR stays nearly flat (0.416 $\rightarrow$ 0.422). The advantage is not universal. AutoHFormer is better on ETTm1 (all extended horizons) and Environment ($H \leq 1440$), Timer-XL on Solar ($H \geq 1440$), and VisionTS on Weather ($H \leq 1440$) and, with Timer-XL, on ETTh2 ($H \leq 1800$).

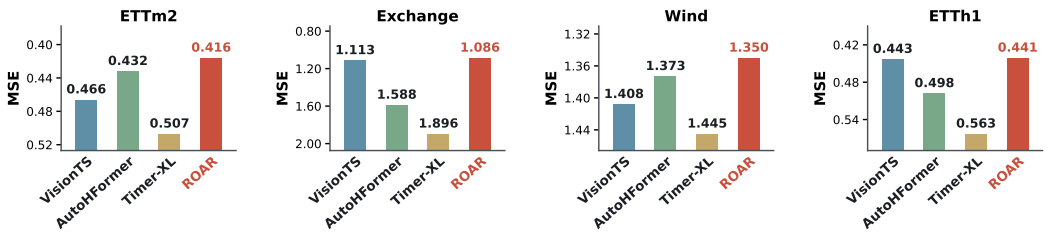

Figure 4: MSE at the extended horizon $H = 1080$ on ETTm2, Exchange, Wind, and ETTh1, the four datasets on which ROAR alone has the lowest MSE at $H = 1080$ (lower is better, and the y axis is inverted and zoomed, so read the printed values). Each bar is the mean over three seeds. The complete comparison is in Table 7.

Table 2: Ablation results. Variant 1 removes random token ordering and trains in raster order, Variant 2 removes the teacher-forcing objective and trains with the free-running branch alone, and Variant 3 removes the free-running objective and trains with teacher forcing alone. We report average results across different prediction horizons, and the best result is marked in **bold**.

Datasets	ETTh1		ETTh2		ETTm1		ETTm2		Exchange		Environment		Wind	
Metrics	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Variant 1	0.398	0.430	0.373	0.409	0.351	0.388	0.278	0.335	0.458	0.460	0.253	0.365	1.140	0.771
Variant 2	0.424	0.450	0.379	0.413	0.357	0.393	0.284	0.339	0.402	0.432	0.249	0.363	1.131	**0.768**
Variant 3	0.402	0.434	0.376	0.414	0.502	0.466	0.364	0.395	**0.328**	**0.392**	0.250	0.368	1.338	0.870
ROAR (full)	**0.392**	**0.426**	**0.362**	**0.402**	**0.348**	**0.385**	**0.276**	**0.333**	**0.328**	**0.392**	**0.247**	**0.360**	**1.130**	**0.768**

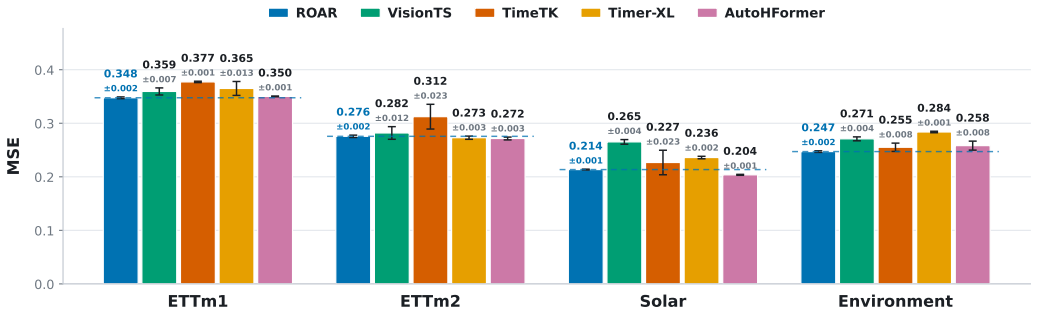

Figure 5: Robustness to random seeds in MSE. Each bar is the four-horizon average over three training seeds and the error bar is one sample standard deviation. The dashed line marks the mean of ROAR. ROAR is compared with the vision-based VisionTS, the numerical TimeTK, and the autoregressive Timer-XL and AutoHFormer on ETTm1, ETTm2, Solar, and Environment. The corresponding MAE figure is Figure 9 in Appendix D.2.

4.5 Model Analyses

4.5.1 Ablation Studies

Table 2 ablates the three components of ROAR on seven datasets. Every variant keeps the tokenizer, predictor, optimizer, and rolling evaluation of the full model, and each cell averages four horizons over three seeds. **Variant 1** removes random token ordering and always serializes tokens in raster order. **Variant 2** removes the teacher-forcing objective and trains with the free-running branch alone. **Variant 3** removes the free-running objective and trains with teacher forcing alone. The full model is best on all seven datasets in both metrics, and the three variants fail in different ways.

(i) Random ordering matters most where the temporal prior is weakest. Variant 1 degrades every dataset. The loss is moderate on the ETT benchmarks (at most 3.0%, on ETTh2) but large

on Exchange, where MSE rises from 0.328 to 0.458 (+40%). Exchange has no stable periodicity, so a fixed raster trajectory carries little useful inductive bias, whereas training over random orders forces the predictor to model dependencies between arbitrary positions of the future canvas. **(ii) Teacher forcing supplies the stable one-step signal.** Variant 2 is never better than the full model, is the worst variant on ETTh1 (+8.2%) and ETTh2 (+4.7%), and also degrades Exchange (+23%). Without teacher forcing, the predictor learns from its own noisy predictions from the first epoch onward. **(iii) The free-running branch drives recursive robustness.** Variant 3 shows the largest degradations in the table, with MSE rising by 44% on ETTm1, 32% on ETTm2, and 18% on Wind, and MAE inflated accordingly. All horizons beyond $H = 96$ are produced by recursive roll-outs, and a predictor trained only with teacher forcing has never consumed its own outputs, so exposure bias compounds across calls. On Exchange, Variant 3 equals the full model because the selected configuration already sets the free-running weight to zero (Appendix B.4). Together with (ii), this result shows that the best balance between the two branches is dataset dependent, which we examine next. In summary, the three components are complementary. Teacher forcing anchors optimization, the free-running branch aligns training with inference, and random ordering enlarges the set of trajectories over which both objectives are optimized.

4.5.2 Hyperparameter Sensitivity Analyses

Having shown that the free-running branch matters, we now ask how sensitive ROAR is to its weight. We sweep the terminal free-running weight λ over $\{0, 0.25, 0.5, 0.75, 1\}$ with three seeds per point and all other settings fixed. Figure 6 shows two regimes. At $\lambda = 0$ the free-running objective is absent, and MSE rises by 44.5% on ETTm1 and 18.5% on Wind, and by 3.9% on ETTh2 and 1.2% on Environment, matching Variant 3 in Table 2. For any $\lambda \in [0.25, 1]$, MSE stays within 2.9% (ETTm1), 1.9% (ETTh2), 1.1% (Environment), and 0.4% (Wind) of the best value, and the seed bands overlap. The shallow minima at $\lambda = 0.25$ on ETTh2 and $\lambda = 0.5$ on Environment lie within one standard deviation of their neighbors and do not indicate a strong preference. What matters is exposing the predictor to its own generated context at all, not the exact mixing ratio, so $\lambda = 0.5$ is a safe default.

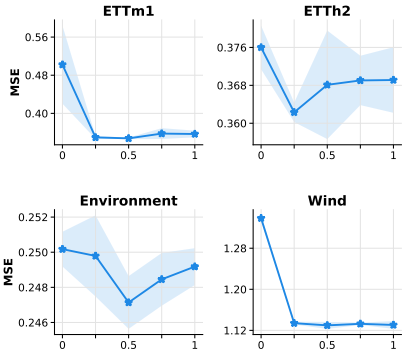

Figure 6: MSE versus the free-running weight λ with all other settings fixed. Points are means over three seeds, and bands show ± 1 standard deviation.

4.5.3 Robustness Analyses

Random ordering injects stochasticity into training, which raises the question of whether the trained forecaster is sensitive to the random seed. Figure 5 reports the mean and standard deviation over three seeds for ROAR and four representative baselines on four datasets; the MAE counterpart is Figure 9 in Appendix D.2 and shows the same picture. ROAR is among the most stable models in every panel. Its MSE standard deviation is at most 0.002 on all four datasets: the smallest on ETTm2 and Solar, and within 0.001 of the smallest on ETTm1 and Environment, where AutoHFormer and Timer-XL are comparably stable. By contrast, VisionTS varies by 0.007 on ETTm1 and 0.012 on ETTm2, and Timer-XL by 0.013 on ETTm1, and TimeTK by 0.023 on both ETTm2 and Solar. On Solar, TimeTK reaches a competitive mean only with a seed spread more than twenty times larger than that of ROAR. This stability does not come at the expense of accuracy, as ROAR has the lowest mean MSE on ETTm1 and Environment and trails the best mean by 0.004 on ETTm2 and 0.010 on Solar. We attribute this behavior to random ordering itself. Optimizing the expected loss over all serialization orders (Equation 2) regularizes the predictor against fitting a single trajectory, and deterministic raster-order inference then reaches nearly the same solution across seeds.

4.5.4 Representation Analysis

The ablation shows that random ordering helps most where the temporal prior is weak; Figure 7 looks at what changes inside the model. We take ROAR and Variant 1 (raster-order training) from Table 2, feed both the same history image of an ETTh1 test window, and compare the four future blocks they generate. ROAR continues the horizontal row structure of the history into the future

region, and the decoded forecast follows the daily oscillation of the target. Variant 1 generates an almost uniformly dark first block that breaks this structure; after inverse rendering, the break appears as a sharp drop at the start of the forecast, and the window MAE rises from 0.539 to 1.435. This observation is consistent with the mechanism proposed in Section 3.2. A raster-order model is only ever trained to draw the first block from the history alone, so an error there propagates to every later block. The random-order model must instead draw each block so that it agrees with the history rows regardless of which blocks precede it, and therefore preserves the periodic structure. The examples in Appendix F show the same pattern on ETTh2, Wind, and Exchange, where Variant 1 draws a future region of the wrong brightness and forecasts a level far from the target, whereas ROAR follows the level shift at the end of the history. These are individual windows rather than test-set averages; the aggregate effect is the Variant 1 row of Table 2.

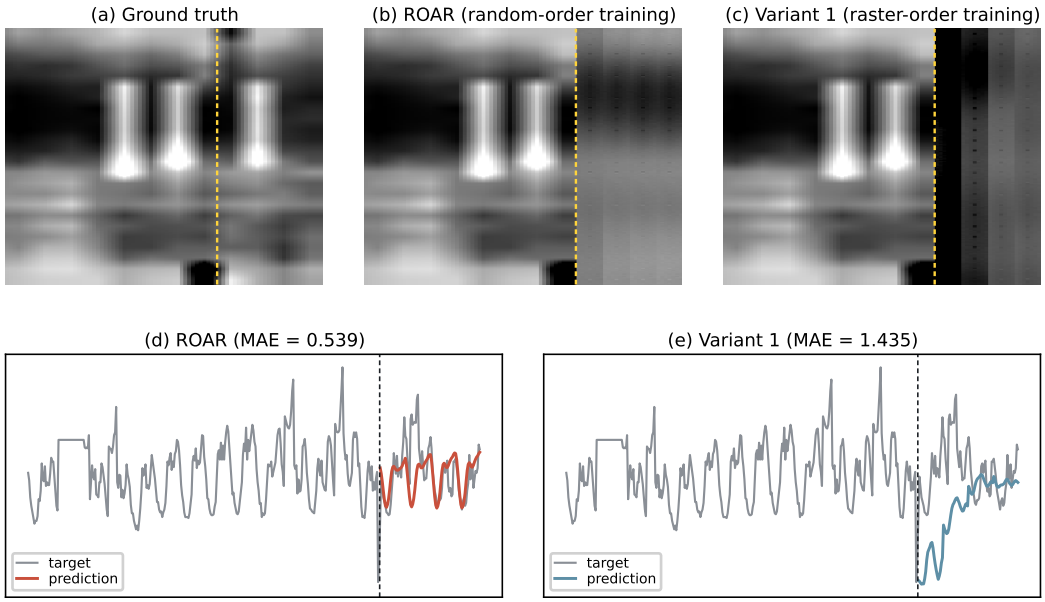

Figure 7: What the predictor draws with and without random ordering. ROAR and Variant 1 (identical except for raster-order training) generate the future region of the same ETTh1 test window (variable LUFL). (a) Ground-truth image, cropped to the last eight days of history and the four future blocks right of the dashed line. (b)–(c) Future regions generated by the two models from the same history. (d)–(e) Forecasts recovered from (b)–(c) by inverse rendering, with each title reporting the MAE for this window.

5 Conclusion and Future Work

This paper presents ROAR, an autoregressive forecaster that generates the endogenous image of a time series in randomly sampled block orders, trained with a dual-branch objective that combines teacher forcing and free running. On eight benchmarks under a shared rolling protocol, one ROAR checkpoint per dataset is competitive with or superior to nine baselines at the standard horizons, ranks first in average error from $H = 336$ onward, and reaches the best average error over the available datasets at every extended horizon up to 2160 steps. Ablations confirm that random ordering, teacher forcing, and the free-running branch each contribute, and visualizations show that random ordering preserves the periodic structure that raster-order training breaks. ROAR still trails the best baselines on the two benchmarks with the most channels and selects periodicity and predictor capacity per dataset. In future work, we plan to model cross-channel interactions within the image representation and to infer periodicity and predictor capacity from the data rather than selecting them per dataset. We also intend to aggregate predictions over multiple generation orders at inference time and to extend the image geometry to irregularly sampled series.

AI USE STATEMENT

We used generative AI tools to edit the manuscript for readability, including correcting grammar, rephrasing sentences, and improving the flow of the text. We did not use generative AI tools to generate data, develop the method or its claims, propose hypotheses, design the experiments, or analyze and interpret the results; theoretical proofs are not applicable to this work. The authors reviewed every AI-edited passage and checked that its technical content and claims were unchanged. We take responsibility for the final content of this work, including text edited with the aid of generative AI.

REFERENCES

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, et al. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024.

Mouxiang Chen, Lefei Shen, Zhuo Li, Xiaoyun Joy Wang, Jianling Sun, and Chenghao Liu. VisionTS: Visual masked autoencoders are free-lunch zero-shot time series forecasters. In *ICML*, 2025.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *ICML*, 2024.

Kuiye Ding, Fanda Fan, Chunyi Hou, Zheya Wang, Lei Wang, Zhengxin Yang, and Jianfeng Zhan. TimeMosaic: Temporal heterogeneity guided time series forecasting via adaptive granularity patch and segment-wise decoding. In *AAAI*, 2026.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.

Jinquan Ji, Yu Cao, Yukun Ma, and Jianzhuo Yan. TITD: Enhancing optimized temporal position encoding with time intervals and temporal decay in irregular time series forecasting. *Applied Intelligence*, 2025.

Yushan Jiang, Kanghui Ning, Zijie Pan, Xuyang Shen, Jingchao Ni, Wenchao Yu, Anderson Schneider, Haifeng Chen, Yuriy Nevmyvaka, and Dongjin Song. Multi-modal time series analysis: A tutorial and survey. In *KDD*, 2025.

Guangyin Jin, Yuxuan Liang, Yuchen Fang, Zezhi Shao, Jincai Huang, Junbo Zhang, and Yu Zheng. Spatio-temporal graph neural networks for predictive learning in urban computing: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 2024a.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. Time-LLM: Time series forecasting by reprogramming large language models. In *ICLR*, 2024b.

Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *ICLR*, 2022.

Hermann Kumbong, Xian Liu, Tsung-Yi Lin, Ming-Yu Liu, Xihui Liu, Ziwei Liu, Daniel Y Fu, Christopher Ré, and David W Romero. HMAR: Efficient hierarchical masked auto-regressive image generation. In *CVPR*, 2025.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long- and short-term temporal patterns with deep neural networks. In *SIGIR*, 2018.

Yan Li, Xinjiang Lu, Yaqing Wang, and Dejing Dou. Generative time series forecasting with diffusion, denoise, and disentanglement. In *NeurIPS*, 2022.

Yuxuan Liang, Haomin Wen, Yuqi Nie, Yushan Jiang, Ming Jin, Dongjin Song, Shirui Pan, and Qingsong Wen. Foundation models for time series analysis: A tutorial and survey. In *KDD*, 2024.

Shengsheng Lin, Weiwei Lin, Wentai Wu, Feiyu Zhao, Ruichao Mo, and Haotong Zhang. SegRNN: Segment recurrent neural network for long-term time-series forecasting. *IEEE Internet of Things Journal*, 2026.

Haoxin Liu, Shangqing Xu, Zhiyuan Zhao, Lingkai Kong, Harshavardhan Kamarthi, Aditya B Sasanur, Megha Sharma, Jiaming Cui, Qingsong Wen, Chao Zhang, and B Aditya Prakash. Time-MMD: Multi-domain multimodal dataset for time series analysis. In *NeurIPS Datasets and Benchmarks Track*, 2024a.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. iTransformer: Inverted transformers are effective for time series forecasting. In *ICLR*, 2024b.

Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. AutoTimes: Autoregressive time series forecasters via large language models. In *NeurIPS*, 2024c.

Yong Liu, Haoran Zhang, Chenyu Li, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer: Generative pre-trained transformers are large time series models. In *ICML*, 2024d.

Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer-XL: Long-context transformers for unified time series forecasting. In *ICLR*, 2025.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *ICLR*, 2023.

Sravani Parvathareddy, Abid Yahya, Lilian Amuhaya, Ravi Samikannu, and Raymond Sogna Suglo. A hybrid machine learning and optimization framework for energy forecasting and management. *Results in Engineering*, 2025.

Ilan Price, Alvaro Sanchez-Gonzalez, Ferran Alet, Tom R Andersson, Andrew El-Kadi, Dominic Masters, Timo Ewalds, Jacklynn Stott, Shakir Mohamed, Peter Battaglia, et al. Probabilistic weather forecasting with machine learning. *Nature*, 2025.

Xiangfei Qiu, Xingjian Wu, Yan Lin, Chenjuan Guo, Jilin Hu, and Bin Yang. DUET: Dual clustering enhanced multivariate time series forecasting. In *KDD*, 2025.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*, 2015.

Weilin Ruan, Siru Zhong, Haomin Wen, and Yuxuan Liang. Vision-enhanced time series forecasting via latent diffusion models. *arXiv preprint arXiv:2502.14887*, 2025.

David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 2020.

ChengAo Shen, Wenchao Yu, Ziming Zhao, Dongjin Song, Wei Cheng, Haifeng Chen, and Jingchao Ni. Multi-modal view enhanced large vision models for long-term time series forecasting. In *NeurIPS*, 2025.

Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen, and Ming Jin. Time-MoE: Billion-scale time series foundation models with mixture of experts. In *ICLR*, 2025.

Yu Shi, Zongliang Fu, Shuo Chen, Bohan Zhao, Wei Xu, Changshui Zhang, and Jian Li. Kronos: A foundation model for the language of financial markets. In *AAAI*, 2026.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. In *NeurIPS*, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Zhang, and Jun Zhou. TimeMixer: Decomposable multiscale mixing for time series forecasting. In *ICLR*, 2024.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Yong Liu, Chen Wang, Mingsheng Long, and Jianmin Wang. Deep time series models: A comprehensive survey and benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.

Ronald J Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1989.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *NeurIPS*, 2021.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. TimesNet: Temporal 2D-variation modeling for general time series analysis. In *ICLR*, 2023.

Jing Xiong, Gongye Liu, Lun Huang, Chengyue Wu, Taiqiang Wu, Yao Mu, Yuan Yao, Hui Shen, Zhongwei Wan, Jinfa Huang, et al. Autoregressive models in vision: A survey. *arXiv preprint arXiv:2411.05902*, 2024.

Fan Zhang, Shiming Fan, and Hua Wang. Time-TK: A multi-offset temporal interaction framework combining transformer and Kolmogorov-Arnold networks for time series forecasting. In *WWW*, 2026a.

Lingzhe Zhang, Tong Jia, Mengxi Jia, Yifan Wu, Aiwei Liu, Yong Yang, Zhonghai Wu, Xuming Hu, Philip Yu, and Ying Li. A survey of AIOps in the era of large language models. *ACM Computing Surveys*, 2025.

Qianru Zhang, Honggang Wen, Ming Li, Dong Huang, Siu-Ming Yiu, Christian S Jensen, and Pietro Liò. Autohformer: Efficient hierarchical autoregressive transformer for time series prediction. In *ICDE*, 2026b.

Siru Zhong, Weilin Ruan, Ming Jin, Huan Li, Qingsong Wen, and Yuxuan Liang. Time-VLM: Exploring multimodal vision-language models for augmented time series forecasting. In *ICML*, 2025.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *AAAI*, 2021.

Appendix (Supplementary Material)

TABLE OF CONTENTS

A Experimental Details **14**

A.1 Benchmarks and Data Processing 14

A.2 Baselines 14

A.3 Implementation Details and Forecasting Protocol 15

B Implementation of the Visual Autoregressive Model **15**

B.1 Conceptual Comparison with MAE-Style Forecasting 15

B.2 History Normalization and Image Geometry 16

B.3 Random-Order Causal Predictor 16

B.4 Teacher-Forced and Free-Running Objectives 16

C Exact Benchmark Configurations **17**

D Detailed Forecasting Results **18**

D.1 Horizon Dependence 18

D.2 Random-Seed Stability 18

D.3 Extended-Horizon Evaluation 19

E Ablation Study **20**

F Additional Representation Visualizations **21**

G Hyperparameter Selection and Result Audit **23**

H Reproducibility Checklist **23**

I Limitations **24**

A Experimental Details

A.1 Benchmarks and Data Processing

The eight benchmarks in Table 1 are ETTm1 and ETTm2 (Zhou et al., 2021), Exchange and Solar (Lai et al., 2018), Weather and Traffic (Wu et al., 2021), Environment (Liu et al., 2024a), and Wind (Li et al., 2022). We also document supplementary ETTh1 and ETTh2 runs. Table 3 records the dimensions, chronological splits, native sampling intervals, and folding periodicities used by the completed runs.

Table 3: Dataset and rendering settings used by the reported ROAR runs. All datasets use a 672-step history, a 96-step training target, and chronological splits. Periodicity is the row length used by the sequence-to-image folding operator.

Dataset	Dim.	History	Split	Frequency	Periodicity	Domain
ETTm1	7	672	6:2:2	15 minutes	96	Electricity
ETTm2	7	672	6:2:2	15 minutes	96	Electricity
ETTh1	7	672	6:2:2	1 hour	24	Electricity
ETTh2	7	672	6:2:2	1 hour	24	Electricity
Exchange	8	672	7:1:2	1 day	24	Economy
Weather	21	672	7:1:2	10 minutes	144	Environment
Solar	137	672	7:1:2	10 minutes	144	Energy
Traffic	862	672	7:1:2	1 hour	24	Transportation
Environment	1	672	7:1:2	1 day	7	Environment
Wind	7	672	7:1:2	15 minutes	96	Energy

Chronological splits and standardization. ETTm1, ETTm2, ETTh1, and ETTh2 use the established 12/4/4-month split implemented by the ETT loader, equivalent to 60/20/20. The remaining datasets use the custom-data loader and a 70/10/20 split; the executed Solar runs also use this split. For every dataset, a separate standard scaler is fitted to each numerical variable using the training interval only and then reused unchanged for validation and test data. The date column is retained for chronological indexing and time-marker construction but is not a forecast target. The current ROAR forward path consumes only the numerical history; it does not use calendar markers.

Window construction without target leakage. Let the target interval of a split be $[b_s, e_s)$, the context length be $L = 672$, and the requested horizon be H. A valid origin t has an input $X_{t-L:t}$ and a target $Y_{t:t+H}$ with the complete target contained in $[b_s, e_s)$. Validation and test inputs may therefore include up to L observations immediately before the split boundary, as is standard for forecasting, while their target timestamps remain inside the requested split. Training windows and all fitted preprocessing statistics are restricted to the training interval.

Channel handling. All experiments use the multivariate forecasting objective, but the ROAR core is channel independent. The ETT loader returns a multivariate window and the model reshapes its C variables into C image instances. The custom-data loader expands its index by the number of channels and returns one variable per item. In both cases, the same visual tokenizer and predictor parameters are shared across variables, and errors are pooled over all variables, origins, and forecast steps. Consequently, ROAR benefits from shared cross-series training but does not explicitly model contemporaneous cross-variable interactions.

A.2 Baselines

The main table compares ROAR with the image-based VisionTS (Chen et al., 2025); the language-guided TimeLLM (Jin et al., 2024b); the autoregressive Timer-XL (Liu et al., 2025) and AutoHFormer (Zhang et al., 2026b); and the numerical or patch-based methods TimeMixer (Wang et al., 2024), TimeTK (Zhang et al., 2026a), iTransformer (Liu et al., 2024b), PatchTST (Nie et al., 2023), and TimesNet (Wu et al., 2023). As specified in the main text, all methods use a 672-step input, a 96-step output window, and recursive rolling for longer horizons.

A.3 Implementation Details and Forecasting Protocol

Each run trains a model with a 672-step history and a 96-step target. The interface also carries a 48-step label segment, but the ROAR wrapper passes only the historical values and the final 96 target values to the visual forecasting core. A single checkpoint is evaluated at $H \in \{96, 192, 336, 720\}$; no horizon-specific model is fitted.

For $H > 96$, the model is called recursively. At rollout step r, the next 96-step block is generated without a future target, appended to the current history, and the most recent L observed-or-generated values become the next input. The numbers of model calls for $H = 96, 192, 336$, and 720 are therefore 1, 2, 4, and 8. When H is not divisible by 96, the final generated block is truncated to the required length. Ground-truth future observations are never fed back.

Each horizon is evaluated on its complete set of valid chronological test windows. The test loader uses `shuffle=False` and `drop_last=False`. Predictions from all batches are concatenated before computing metrics, so a smaller final batch contributes in proportion to its number of targets. With N_H scalar targets after pooling origins, steps, and variables, we report

$$\mathrm{MSE}_H = \frac{1}{N_H}\sum_{i=1}^{N_H}(y_i - \hat{y}_i)^2, \qquad \mathrm{MAE}_H = \frac{1}{N_H}\sum_{i=1}^{N_H}|y_i - \hat{y}_i|. \tag{5}$$

Evaluation uses `inverse=False`; hence the reported errors are on the training-standardized scale, matching the main table. The Average column is the unweighted arithmetic mean of the four horizon metrics.

B Implementation of the Visual Autoregressive Model

B.1 Conceptual Comparison with MAE-Style Forecasting

Figure 8 contrasts the prediction structures of MAE-style visual forecasting and ROAR. The former predicts future tokens in a fixed order, whereas ROAR samples serialization orders while retaining each token's physical position.

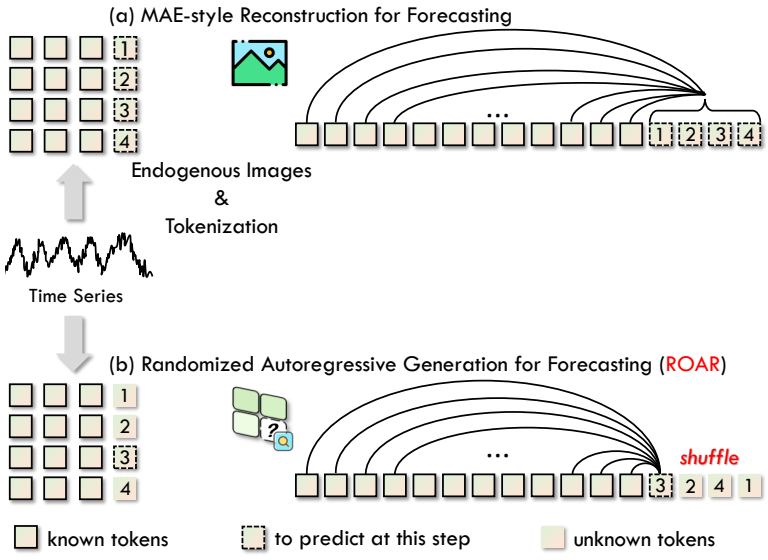

Figure 8: Conceptual comparison between (a) MAE-style reconstruction with a predefined visual-token prediction pattern and (b) ROAR training with sampled autoregressive serialization orders.

B.2 History Normalization and Image Geometry

ROAR applies a second, sample-wise normalization after dataset standardization. For each sample and variable, the mean and variance use only the L observed values:

$$\mu = \frac{1}{L}\sum_{t=1}^{L} x_t, \qquad s = \frac{\sqrt{\mathrm{Var}(X) + 10^{-5}}}{0.4}, \qquad \widetilde{X} = \frac{X - \mu}{s}. \tag{6}$$

Both μ and s are detached from gradient computation. Predictions are mapped back through $\hat{Y} = s\widetilde{Y} + \mu$ before the benchmark loss is formed. Dataset standardization therefore defines the reported metric scale, whereas sample-wise normalization controls the range presented to the visual modules; neither stage uses future statistics.

For each variable, the normalized history is folded by its dominant period P into a two-dimensional matrix:

$$X_{\mathrm{2D}}[i, j] = \widetilde{X}[(j-1)P + i], \tag{7}$$

so the j-th period occupies the j-th column and observations at the same phase of different periods share a row. For example, with hourly observations and $P = 24$, each column represents one day and each row represents a fixed hour. Replicate padding is added only when the history or target length is not divisible by P.

Bilinear interpolation maps the folded history to a visible image region, and a zero-valued future region is appended on the right. For the 672-step-history runs, these regions have sizes 224×196 and 224×28; for ETTh1 and ETTh2, they have sizes 224×192 and 224×32. Both layouts produce a 224×224 endogenous image. Repeating the grayscale image over three channels provides the input shape expected by the visual modules without introducing additional information.

Token geometry. A frozen history encoder maps the visible region to six or seven prefix embeddings, depending on the tokenizer configuration. The future tokenizer divides the future strip into four contiguous blocks and maps them to four future latent vectors; the paired decoder maps these latents back to the future strip. These internal visual blocks do not change the 672-step context or the 96-step forecasting target.

B.3 Random-Order Causal Predictor

The predictor is a pre-norm Transformer encoder used with a causal mask. Each layer has multi-head self-attention, a GELU feed-forward network of width $4d$, residual connections, and the dataset-specific dropout in Table 4. Learned positional embeddings encode physical token identity. If a history or future permutation is sampled, the embedding and its physical-position index are permuted together; random serialization therefore changes the factorization order without discarding spatial identity.

During training, every example receives independently sampled history and future-block permutations, matching the random-order predictor in the main text. The causal mask prevents each serialized position from attending to any later position in the sampled order. At evaluation, the history order is a fixed seeded permutation (seed 0) and the four future blocks are generated in left-to-right raster order. Evaluation is consequently deterministic for a fixed checkpoint.

B.4 Teacher-Forced and Free-Running Objectives

In the teacher-forced branch, the future latents are shifted by one position and the predictor receives the ground-truth latent preceding each current target in the sampled order. In the free-running branch, predicted latents are fed back autoregressively. Both branches decode the predicted latents to the future image and invert the image-to-series mapping. Writing each branch loss in its mean-squared form, the implemented objective is the following specialization of Equation 4 in the main text:

$$\mathcal{L}_e = \lambda_{\mathrm{TF}}\mathcal{L}_{\mathrm{TF}} + \lambda_{\mathrm{FR},e}\mathcal{L}_{\mathrm{FR}}, \qquad \lambda_{\mathrm{TF}} = 1, \quad \lambda_{\mathrm{FR},e} = \lambda \min\left(1, \frac{e+1}{W_{\mathrm{FR}}}\right), \tag{8}$$

Table 4: Exact configurations associated with the main-table ROAR entries. The tokenizer and predictor columns give hidden width and layer count (d, l) for the frozen column tokenizer and trainable autoregressive predictor. FR gives the terminal free-running loss weight and its warm-up length in epochs.

Dataset	Tok. (d, l)	Pred. (d, l)	Heads	Dropout	Learning rate	Batch	Epochs/patience	FR weight/warm-up
ETTh1	$(128, 2)$	$(256, 4)$	4	0.00	1.0×10^{-3}	128	100/10	0.25/8
ETTh2	$(256, 3)$	$(256, 5)$	8	0.00	3.75×10^{-4}	32	20/5	0.25/8
ETTm1	$(256, 3)$	$(128, 1)$	8	0.10	9.0×10^{-4}	64	100/10	0.50/3
ETTm2	$(256, 3)$	$(128, 2)$	8	0.30	1.0×10^{-3}	128	100/10	0.25/8
Exchange	$(256, 3)$	$(256, 3)$	8	0.00	5.0×10^{-4}	128	100/10	0.00/0
Weather	$(128, 2)$	$(512, 3)$	8	0.10	6.0×10^{-4}	64	20/5	0.25/8
Solar	$(256, 3)$	$(512, 4)$	8	0.00	2.5×10^{-4}	1024	20/5	0.25/5
Traffic	$(256, 3)$	$(768, 4)$	8	0.00	1.0×10^{-4}	2048	20/5	0.50/5
Environment	$(384, 3)$	$(1024, 4)$	8	0.30	1.5×10^{-4}	64	100/10	0.50/5
Wind	$(128, 2)$	$(128, 2)$	8	0.05	1.0×10^{-4}	64	100/10	0.50/5

where the common factor $1/F$ relating MSE to the squared ℓ_2 terms in Equation 4 does not change the optimizer, and λ and W_{FR} are reported in Table 4. Exchange uses $\lambda = 0$; its final training objective uses only the teacher-forced loss. This dataset-specific setting is preserved in the corresponding ablations.

The history tokenizer, future tokenizer, and future decoder are loaded from their selected checkpoints, set to evaluation mode, and frozen. Only the autoregressive predictor is optimized with Adam to minimize MSE. Early stopping keeps the checkpoint with the smallest validation loss using the patience listed in Table 4. Python, NumPy, PyTorch, CUDA, data-loader, and serialization randomness are initialized from the run seed.

C Exact Benchmark Configurations

Table 4 transcribes the executed configurations rather than nominal defaults. All rows use a 224×224 image, bilinear interpolation, eight attention heads (four for ETTh1), six or seven history tokens, four future latents, random training order, raster future evaluation order, and a frozen continuous column tokenizer. Predictor capacity and optimization settings are dataset specific because the number of training windows, variables, and signal regularity differ substantially.

Batch size refers to items returned by the corresponding loader. For ETT, an item contains all seven variables before the model flattens channels. For custom datasets, an item contains one variable. This distinction explains why large nominal batches for Solar and Traffic remain tractable and prevents a misleading comparison of batch size across loaders.

Table 5: Horizon-wise ROAR results. Values are measured on the training-standardized scale. Each entry is the mean ± sample standard deviation over three seeds.

Dataset (n)	Metric	$H = 96$	$H = 192$	$H = 336$	$H = 720$	Average
ETTm1 (3)	MSE	0.2886±0.0031	0.3286±0.0025	0.3593±0.0024	0.4139±0.0031	0.3476±0.0016
	MAE	0.3476±0.0012	0.3728±0.0006	0.3935±0.0003	0.4242±0.0009	0.3845±0.0002
ETTm2 (3)	MSE	0.1921±0.0012	0.2435±0.0028	0.2922±0.0029	0.3747±0.0025	0.2756±0.0022
	MAE	0.2805±0.0009	0.3129±0.0016	0.3436±0.0021	0.3941±0.0034	0.3328±0.0020
ETTh1 (3)	MSE	0.3660±0.0055	0.3881±0.0038	0.3997±0.0038	0.4130±0.0067	0.3917±0.0045
	MAE	0.4061±0.0044	0.4221±0.0032	0.4309±0.0027	0.4462±0.0051	0.4263±0.0033
ETTh2 (3)	MSE	0.2984±0.0040	0.3559±0.0021	0.3820±0.0021	0.4130±0.0114	0.3623±0.0021
	MAE	0.3570±0.0022	0.3927±0.0005	0.4154±0.0026	0.4440±0.0080	0.4023±0.0021
Exchange (3)	MSE	0.0900±0.0027	0.1845±0.0047	0.3341±0.0114	0.7016±0.0904	0.3275±0.0260
	MAE	0.2148±0.0023	0.3110±0.0038	0.4207±0.0076	0.6223±0.0516	0.3922±0.0153
Weather (3)	MSE	0.1717±0.0016	0.2121±0.0020	0.2588±0.0028	0.3256±0.0040	0.2421±0.0020
	MAE	0.2289±0.0012	0.2632±0.0016	0.2980±0.0022	0.3430±0.0031	0.2833±0.0018
Solar (3)	MSE	0.1915±0.0032	0.2074±0.0024	0.2192±0.0013	0.2360±0.0026	0.2135±0.0008
	MAE	0.2442±0.0012	0.2559±0.0022	0.2661±0.0035	0.2808±0.0058	0.2617±0.0029
Traffic (3)	MSE	0.3898±0.0018	0.4068±0.0020	0.4238±0.0024	0.4653±0.0028	0.4214±0.0022
	MAE	0.2771±0.0011	0.2838±0.0010	0.2913±0.0010	0.3128±0.0011	0.2913±0.0010
Environment (3)	MSE	0.2578±0.0017	0.2508±0.0017	0.2414±0.0015	0.2386±0.0011	0.2471±0.0015
	MAE	0.3651±0.0019	0.3618±0.0020	0.3580±0.0019	0.3569±0.0027	0.3605±0.0017
Wind (3)	MSE	0.9017±0.0056	1.0692±0.0087	1.2082±0.0088	1.3399±0.0057	1.1298±0.0068
	MAE	0.6573±0.0037	0.7441±0.0044	0.8073±0.0041	0.8617±0.0031	0.7676±0.0038

D Detailed Forecasting Results

Table 6 organizes forecasting results by dataset, model, and prediction horizon. Each horizon has separate MSE and MAE columns. The Average entries for the eight datasets reproduce the main table.

The repeated-seed ROAR results in Table 5 use identical architecture, optimization, tokenizer checkpoints, curricula, and evaluation settings within each seed group.

D.1 Horizon Dependence

From $H = 96$ to $H = 720$, MSE changes by $+43.4\%$ on ETTm1, $+95.1\%$ on ETTm2, $+12.8\%$ on ETTh1, $+38.4\%$ on ETTh2, $+679.6\%$ on Exchange, $+89.6\%$ on Weather, $+23.2\%$ on Solar, $+19.4\%$ on Traffic, -7.4% on Environment, and $+48.6\%$ on Wind. Exchange is the most challenging case: at $H = 720$, it has both the largest relative degradation and the largest seed standard deviation. Traffic and Solar show much milder relative growth.

Environment is the exception to monotonic degradation. Its MSE decreases from 0.2578 to 0.2386 as the evaluated horizon grows. Because each horizon has its own set of valid forecast origins, this observation should not be interpreted as evidence that recursion intrinsically improves accuracy. It indicates that the daily Environment test segments evaluated at longer horizons are, on average, easier under the present split. We therefore report all horizon values rather than fitting or claiming a universal error-growth law.

D.2 Random-Seed Stability

All eight main-table datasets have repeated runs with an exact configuration match. Supplementary ETTh1 uses seeds 2021, 2023, and 2025, and ETTh2 uses seeds 2023, 2031, and 2038; ETTm1, ETTm2, Exchange, Weather, Solar, Traffic, Environment, and Wind use three seeds (2021, 2023, and 2025). The average-MSE sample standard deviation is largest for Exchange (0.0260), followed by Wind (0.0068) and ETTh1 (0.0045); the remaining repeated groups are below 0.003. Every reported group has $n = 3$; the seed identities are listed explicitly because ETTh2 uses a different triplet.

Figure 9 is the MAE counterpart of Figure 5 in the main text, on the same four datasets and the same five models. The ordering of the seed spreads matches the MSE panels: ROAR stays below 0.003 everywhere, whereas VisionTS reaches 0.008 on ETTm2 and TimeTK reaches 0.017.

Table 6: Full-shot forecasting results by dataset and prediction horizon.

Dataset	H	**ROAR**		VisionTS		TimeLLM		TimeMixer		TimeTK		Timer-XL		AutoHFormer		iTransformer		PatchTST		TimesNet	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	0.2886	0.3476	**0.2879**	**0.3327**	0.3004	0.3575	0.320	0.366	0.3061	0.3563	0.292	0.348	0.2923	0.3520	0.319	0.372	0.294	0.354	0.338	0.375
	192	**0.3286**	0.3728	0.3320	**0.3612**	0.3435	0.3838	0.367	0.394	0.3525	0.3839	0.337	0.377	0.3303	0.3756	0.358	0.395	0.335	0.380	0.371	0.387
	336	**0.3593**	0.3935	0.3691	**0.3868**	0.3771	0.4063	0.404	0.416	0.3880	0.4060	0.376	0.401	0.3618	0.3945	0.394	0.416	0.372	0.402	0.410	0.411
	720	**0.4139**	0.4242	0.4487	0.4323	0.4419	0.4442	0.474	0.454	0.4622	0.4470	0.455	0.444	0.4162	**0.4223**	0.465	0.453	0.429	0.435	0.478	0.450
	Avg	**0.348**	0.385	0.359	**0.378**	0.366	0.398	0.391	0.408	0.377	0.398	0.365	0.393	0.350	0.386	0.384	0.409	0.358	0.393	0.399	0.406
ETTm2	96	0.1921	0.2805	**0.1749**	0.2626	0.1817	0.2691	0.181	0.267	0.2352	0.3186	0.176	**0.258**	0.1767	0.2667	0.185	0.275	0.177	0.266	0.187	0.267
	192	0.2435	0.3129	0.2407	0.3085	0.2396	0.3070	0.245	0.310	0.2815	0.3466	0.244	**0.299**	**0.2351**	0.3066	0.247	0.316	0.239	0.308	0.249	0.309
	336	0.2922	0.3436	0.3068	0.3520	0.3012	0.3454	0.304	0.351	0.3277	0.3730	0.291	**0.338**	**0.2902**	0.3431	0.302	0.353	0.291	0.345	0.321	0.351
	720	**0.3747**	**0.3941**	0.4050	0.4141	0.4028	0.4042	0.424	0.425	0.4049	0.4172	0.381	0.397	0.3841	0.4010	0.404	0.415	0.383	0.404	0.497	0.403
	Avg	0.276	0.333	0.282	0.334	0.281	0.331	0.289	0.338	0.312	0.364	0.273	**0.323**	**0.272**	0.329	0.285	0.340	0.273	0.331	0.314	0.333
Exchange	96	**0.0900**	0.2148	0.0918	**0.2144**	0.1257	0.2546	0.1139	0.2476	0.1008	0.2285	0.1620	0.2818	0.1053	0.2332	0.1206	0.2501	0.0956	0.2191	0.2498	0.3749
	192	**0.1845**	**0.3110**	0.1937	0.3180	0.2303	0.3468	0.2445	0.3632	0.1917	0.3157	0.3515	0.4237	0.2000	0.3232	0.2202	0.3359	0.2001	0.3214	0.4226	0.4780
	336	**0.3341**	**0.4207**	0.3708	0.4469	0.3969	0.4644	0.4849	0.5152	0.3542	0.4354	0.7156	0.6218	0.3773	0.4492	0.4152	0.4631	0.3782	0.4509	0.7914	0.6396
	720	**0.7016**	**0.6223**	0.8522	0.6991	1.0130	0.7698	1.1250	0.8179	0.7956	0.6622	2.1745	1.1181	1.0521	0.7827	0.9308	0.7205	0.9700	0.7447	2.5084	1.1422
	Avg	**0.328**	**0.392**	0.377	0.420	0.441	0.459	0.492	0.486	0.361	0.410	0.851	0.611	0.434	0.447	0.422	0.442	0.411	0.434	0.993	0.659
Weather	96	0.1717	0.2289	**0.1413**	**0.1881**	0.1647	0.2187	0.149	0.203	0.1589	0.2129	0.1751	0.2274	0.1534	0.2045	0.170	0.223	0.149	0.202	0.169	0.228
	192	0.2121	0.2632	**0.1869**	**0.2319**	0.2087	0.2585	0.197	0.247	0.2051	0.2549	0.2272	0.2714	0.2011	0.2484	0.218	0.263	0.196	0.247	0.222	0.269
	336	0.2588	0.2980	**0.2390**	**0.2731**	0.2581	0.2980	0.247	0.285	0.2549	0.2930	0.2832	0.3125	0.2534	0.2889	0.274	0.302	0.249	0.287	0.290	0.310
	720	0.3256	0.3430	**0.3127**	**0.3241**	0.3286	0.3502	0.319	0.335	0.3276	0.3435	0.3473	0.3563	0.3267	0.3423	0.352	0.354	0.321	0.339	0.376	0.364
	Avg	0.242	0.283	**0.220**	**0.254**	0.240	0.281	0.228	0.268	0.237	0.276	0.258	0.292	0.234	0.271	0.254	0.286	0.229	0.269	0.264	0.293
Solar	96	0.1915	0.2442	0.2241	0.2256	0.1961	0.2698	0.1794	0.2426	0.1971	0.2366	0.1989	**0.2116**	**0.1711**	0.2406	0.1874	0.2606	0.1744	0.2324	0.1800	0.2720
	192	0.2074	0.2559	0.2574	0.2468	0.2222	0.2926	0.1981	0.2567	0.2188	0.2541	0.2283	**0.2328**	**0.1939**	0.2587	0.2080	0.2750	0.1955	0.2493	0.1990	0.2860
	336	0.2192	0.2661	0.2805	0.2621	0.2459	0.3133	0.2147	0.2700	0.2346	0.2667	0.2520	**0.2484**	**0.2117**	0.2729	0.2245	0.2859	0.2155	0.2644	0.2200	0.3010
	720	**0.2360**	0.2808	0.2987	0.2776	0.2892	0.3534	0.2462	0.2916	0.2562	0.2804	0.2647	**0.2586**	0.2379	0.2902	0.2406	0.2963	0.2489	0.2869	0.2510	0.3210
	Avg	0.214	0.262	0.265	0.253	0.238	0.307	0.210	0.265	0.227	0.259	0.236	**0.238**	**0.204**	0.266	0.215	0.279	0.209	0.258	0.213	0.295
Traffic	96	0.3898	0.2771	0.3676	0.2411	0.3822	0.2931	0.360	0.255	0.4856	0.3600	0.3736	**0.2341**	0.3999	0.2982	**0.351**	0.257	0.359	0.255	0.593	0.315
	192	0.4068	0.2838	0.3908	0.2520	0.4029	0.3097	0.380	0.263	0.4933	0.3619	0.3939	**0.2455**	0.4084	0.3009	**0.374**	0.266	0.377	0.265	0.596	0.317
	336	0.4238	0.2913	0.4100	0.2630	0.4285	0.3312	0.396	0.272	0.5041	0.3652	0.4107	**0.2560**	0.4221	0.3080	**0.392**	0.274	0.393	0.276	0.600	0.319
	720	0.4653	0.3128	0.4565	0.2891	0.4994	0.3815	0.435	0.296	0.5399	0.3786	0.4508	**0.2807**	0.4607	0.3280	**0.433**	0.291	0.436	0.305	0.619	0.335
	Avg	0.421	0.291	0.406	0.261	0.428	0.329	0.393	0.272	0.506	0.366	0.407	**0.254**	0.423	0.309	**0.388**	0.272	0.391	0.275	0.602	0.322
Environment	96	**0.2578**	0.3651	0.2603	**0.3626**	0.2825	0.3782	0.2610	0.3702	0.2659	0.3712	0.2846	0.3885	0.2697	0.3665	0.3071	0.4125	0.2808	0.3897	0.2688	0.3690
	192	**0.2508**	**0.3618**	0.2582	0.3710	0.2733	0.3742	0.2554	0.3688	0.2589	0.3686	0.2985	0.4036	0.2618	0.3628	0.3076	0.4149	0.2726	0.3857	0.2626	0.3676
	336	**0.2414**	**0.3580**	0.2709	0.3952	0.2625	0.3692	0.2484	0.3681	0.2497	0.3649	0.2814	0.3915	0.2540	0.3594	0.2988	0.4097	0.2630	0.3823	0.2523	0.3632
	720	**0.2386**	0.3569	0.2942	0.4221	0.2557	0.3652	0.2504	0.3740	0.2461	0.3647	0.2703	0.3849	0.2473	**0.3566**	0.3063	0.4171	0.2547	0.3794	0.2489	0.3635
	Avg	**0.247**	**0.360**	0.271	0.388	0.268	0.372	0.254	0.370	0.255	0.367	0.284	0.392	0.258	0.361	0.305	0.414	0.268	0.384	0.258	0.366
Wind	96	0.9017	0.6573	**0.8908**	**0.6418**	0.9408	0.6863	1.1508	0.7320	0.9288	0.6619	0.9959	0.6872	0.9447	0.6747	1.2385	0.7707	0.9003	0.6641	1.0703	0.7109
	192	**1.0692**	0.7441	1.0786	**0.7402**	1.1041	0.7630	1.3643	0.8286	1.0929	0.7465	1.2211	0.7897	1.1220	0.7599	1.4775	0.8687	1.0862	0.7538	1.2521	0.7892
	336	**1.2082**	**0.8073**	1.2350	0.8151	1.2462	0.8232	1.4848	0.8835	1.2297	0.8092	1.3664	0.8527	1.2590	0.8194	1.6044	0.9205	1.2277	0.8160	1.3993	0.8486
	720	**1.3399**	**0.8617**	1.3978	0.8831	1.3642	0.8702	1.5492	0.9174	1.3604	0.8646	1.4563	0.8921	1.3659	0.8649	1.6894	0.9572	1.3451	0.8654	1.4971	0.8889
	Avg	**1.130**	**0.768**	1.151	0.770	1.164	0.786	1.387	0.840	1.153	0.771	1.260	0.805	1.173	0.780	1.502	0.879	1.140	0.775	1.305	0.809
1st Count		**24**		**18**		**0**		**0**		**0**		**11**		**7**		**4**		**0**		**0**	

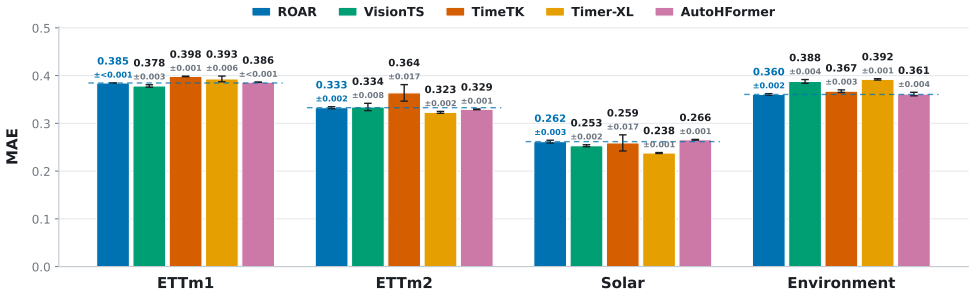

Figure 9: Robustness to random seeds in MAE, the counterpart of Figure 5. Each bar is the four-horizon average over three training seeds and the error bar is one sample standard deviation; the dashed line marks the mean of ROAR.

D.3 Extended-Horizon Evaluation

Table 7 evaluates the same checkpoints at $H \in \{1080, 1440, 1800, 2160\}$, corresponding to 12, 15, 19, and 23 recursive 96-step calls. No ground-truth future block is introduced between calls. Exchange has valid test windows through 1440 but not through 1800 or 2160. Among the nine datasets in Table 7, ROAR obtains lower MSE than VisionTS on seven datasets at both 1080 and

Table 7: Ultra-long-horizon results. Best and second-best results are marked in **bold** and underlined, respectively.

Datasets	Horizon	ROAR (**Ours**)		VisionTS (2025)		Timer-XL (2025)		AutoHFormer (2026b)	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	1080	**0.441**	0.468	0.443	**0.460**	0.563	0.533	0.498	0.506
	1440	**0.476**	0.493	0.484	**0.488**	0.603	0.573	0.564	0.546
	1800	**0.527**	0.528	0.531	**0.519**	0.735	0.627	0.657	0.595
	2160	**0.582**	0.566	0.584	**0.553**	0.848	0.685	0.758	0.647
ETTh2	1080	0.509	0.501	**0.495**	**0.496**	0.507	0.503	0.534	0.522
	1440	0.574	0.533	0.570	0.531	**0.543**	**0.525**	0.608	0.559
	1800	0.604	**0.547**	0.615	0.550	**0.590**	0.550	0.665	0.585
	2160	**0.619**	**0.555**	0.671	0.575	0.643	0.576	0.686	0.596
ETTm1	1080	0.447	0.441	0.466	0.443	0.497	0.474	**0.446**	**0.438**
	1440	0.468	0.450	0.490	0.458	0.514	0.483	**0.465**	**0.448**
	1800	0.485	0.457	0.509	0.470	0.528	0.490	**0.482**	**0.456**
	2160	0.492	0.461	0.516	0.480	0.532	0.496	**0.482**	**0.460**
ETTm2	1080	**0.416**	**0.425**	0.466	0.455	0.507	0.503	0.432	0.434
	1440	**0.425**	**0.436**	0.475	0.467	0.543	0.524	0.445	0.448
	1800	**0.419**	**0.437**	0.468	0.468	0.590	0.550	0.445	0.453
	2160	**0.422**	**0.440**	0.471	0.471	0.643	0.576	0.451	0.458
Exchange	1080	**1.086**	**0.749**	1.113	0.767	1.896	1.004	1.588	0.943
	1440	**2.163**	**1.100**	2.294	1.147	2.659	1.213	2.658	1.237
	1800	–	–	–	–	–	–	–	–
	2160	–	–	–	–	–	–	–	–
Weather	1080	0.356	0.363	**0.349**	**0.350**	0.379	0.376	0.371	0.371
	1440	0.377	0.375	**0.375**	**0.366**	0.400	0.391	0.400	0.388
	1800	**0.394**	0.386	0.396	**0.381**	0.416	0.402	0.423	0.405
	2160	**0.406**	0.394	0.412	**0.392**	0.428	0.410	0.441	0.417
Solar	1080	**0.253**	0.293	0.308	**0.286**	**0.253**	0.290	0.259	0.302
	1440	0.273	0.305	0.315	**0.293**	**0.259**	0.297	0.288	0.314
	1800	0.293	0.316	0.323	**0.299**	**0.263**	0.302	0.318	0.327
	2160	0.317	0.327	0.335	**0.306**	**0.274**	0.310	0.352	0.340
Environment	1080	0.240	**0.359**	0.303	0.437	0.268	0.385	**0.238**	0.363
	1440	0.244	**0.364**	0.315	0.451	0.269	0.387	**0.242**	0.370
	1800	**0.248**	**0.368**	0.320	0.457	0.267	0.386	**0.248**	0.376
	2160	**0.252**	**0.371**	0.326	0.464	0.267	0.386	0.255	0.384
Wind	1080	**1.350**	**0.867**	1.408	0.889	1.445	0.892	1.373	0.870
	1440	**1.326**	**0.860**	1.384	0.885	1.384	0.876	1.340	0.861
	1800	**1.296**	**0.853**	1.353	0.878	1.354	0.870	1.314	0.856
	2160	**1.304**	**0.859**	1.357	0.884	1.366	0.879	1.320	0.862
Average	1080	**0.566**	**0.496**	0.595	0.509	0.702	0.551	0.638	0.528
	1440	**0.703**	**0.546**	0.745	0.565	0.797	0.585	0.779	0.575
	1800	**0.533**	**0.487**	0.564	0.503	0.593	0.522	0.569	0.507
	2160	**0.549**	**0.497**	0.584	0.516	0.625	0.540	0.593	0.521

1440. Excluding unavailable Exchange entries, ROAR obtains lower MSE on all eight datasets at 1800 and 2160. ETTh2 and Weather are the exceptions at the two shorter extended horizons.

E Ablation Study

Tables 8 and 9 summarize the same seven-dataset, three-seed ablation comparison as Table 2 in the main text. Variant 1 replaces independently sampled permutations with fixed raster order. Variant 2 removes teacher forcing and uses only the free-running branch. Variant 3 removes the free-running branch by setting its weight to zero. Every comparison keeps the dataset-specific tokenizer, predictor, optimizer, epoch budget, and evaluation procedure fixed. Figure 10 visualizes the resulting relative MSE changes.

Table 8: Controlled ablation definitions. Each variant changes only the listed training mechanism; architecture, optimizer, data split, and rolling evaluation are unchanged.

Model	Random block order	Teacher forcing	Free-running branch
ROAR (full)	Yes	Yes	Dataset-specific
Variant 1	No	Yes	Dataset-specific
Variant 2	Yes	No	Weight 1.0
Variant 3	Yes	Yes	Weight 0.0

Table 9: Seven-dataset ablation summary corresponding to Table 2. Each cell is the three-seed mean MSE averaged over the four prediction horizons. Δ is the macro-average relative MSE change against the full model across the seven datasets; positive values indicate degradation.

Model	ETTh1	ETTh2	ETTm1	ETTm2	Exchange	Environment	Wind	Δ
ROAR (full)	**0.392**	**0.362**	**0.348**	**0.276**	**0.328**	**0.247**	**1.130**	0.0%
Variant 1	0.398	0.373	0.351	0.278	0.458	0.253	1.140	+7.0%
Variant 2	0.424	0.379	0.357	0.284	0.402	0.249	1.131	+6.0%
Variant 3	0.402	0.376	0.502	0.364	**0.328**	0.250	1.338	+14.6%

Random ordering. Removing random ordering increases macro-average relative MSE by 7.0%. The effect is concentrated on Exchange, where MSE rises by 39.6%, followed by ETTh2 at 3.0%; the increases on the other five datasets range from 0.7% to 2.4%.

Teacher forcing. Free-running-only training raises macro-average MSE by 6.0%. The largest degradation is on Exchange (22.6%), followed by ETTh1 (8.2%) and ETTh2 (4.7%). Teacher forcing supplies a stable one-step latent target, but its importance depends on signal and configuration.

Free-running objective. Teacher-forcing-only training has the largest macro-average degradation, 14.6%. It raises MSE by 44.3% on ETTm1, 31.9% on ETTm2, and 18.4% on Wind. Exchange is unchanged by construction because its selected full configuration already has zero free-running weight. Environment changes by only 1.2%. These results support exposure to generated latents for recursive robustness while also showing why the loss weight was selected per dataset.

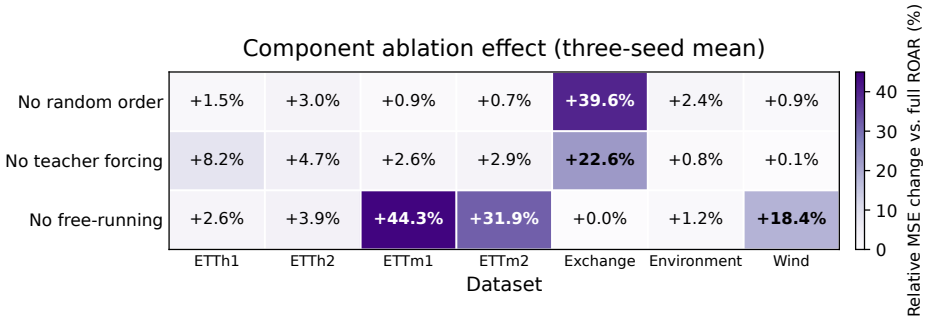

Figure 10: Relative four-horizon average-MSE change of each ablation against full ROAR. Values are computed from the three-seed means in Table 9; positive values denote degradation.

F Additional Representation Visualizations

Figures 11–13 extend Figure 7 to three more datasets. Each figure uses the ROAR and Variant 1 checkpoints of the same seed from the ablation study and shows a single 96-step generation, i.e., the four future blocks of the image. As in the main text, the history image is the same for both models, and the examples illustrate individual windows rather than test-set averages.

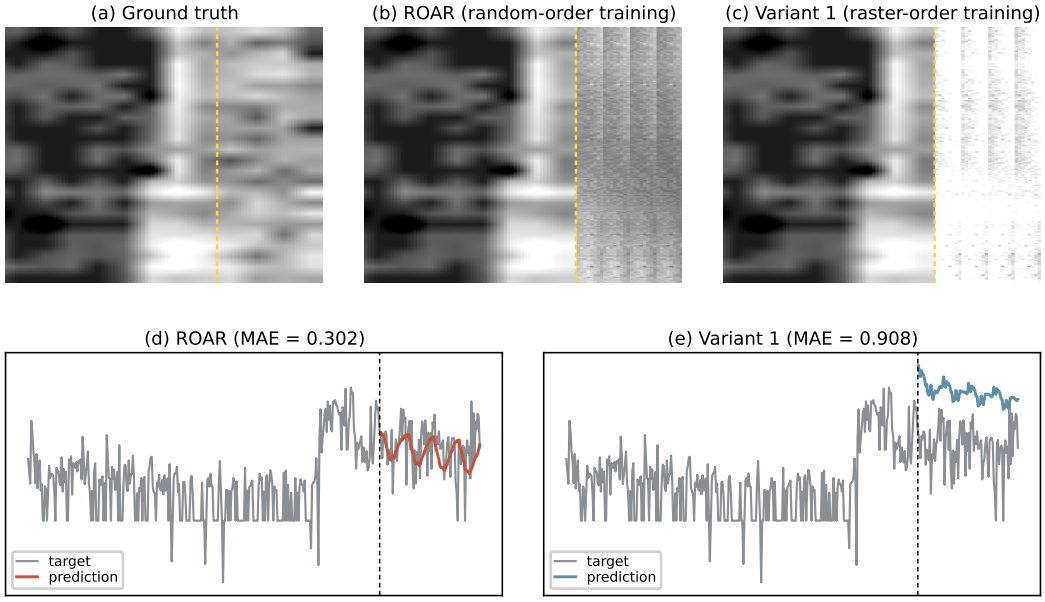

Figure 11: ROAR and Variant 1 on an ETTh2 test window (variable MULL). Images are cropped to the last eight days of history and the four future blocks (right of the dashed line). Variant 1 generates a nearly white future region, and its forecast is shifted far above the target.

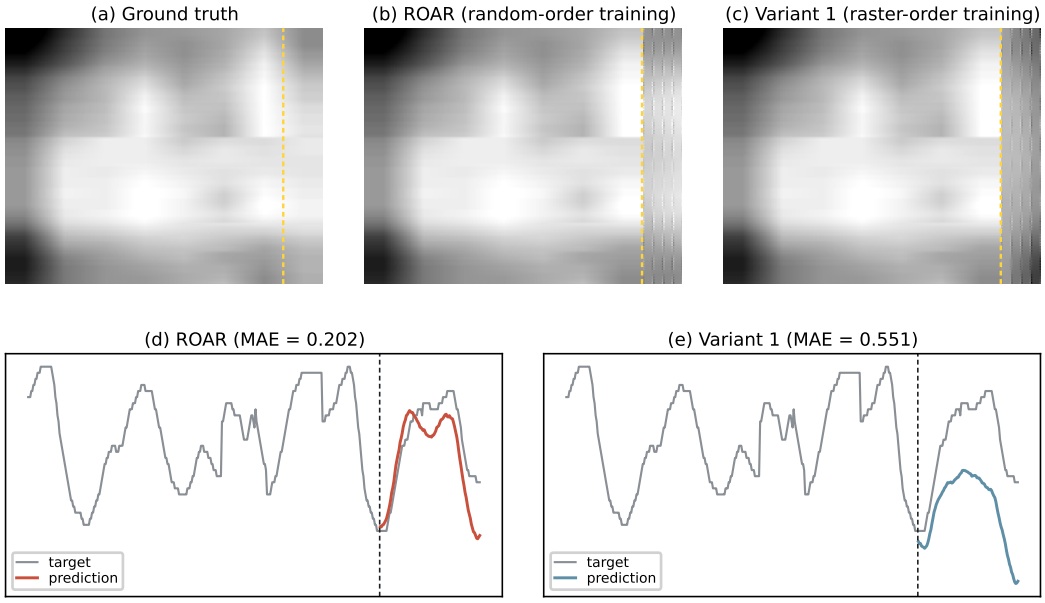

Figure 12: ROAR and Variant 1 on a Wind test window (fifth variable). Wind is folded with a period of 96 steps, so the full image is shown and the four future blocks occupy the narrow region to the right of the dashed line. Variant 1 draws a darker future region, and its forecast underestimates the level of the target.

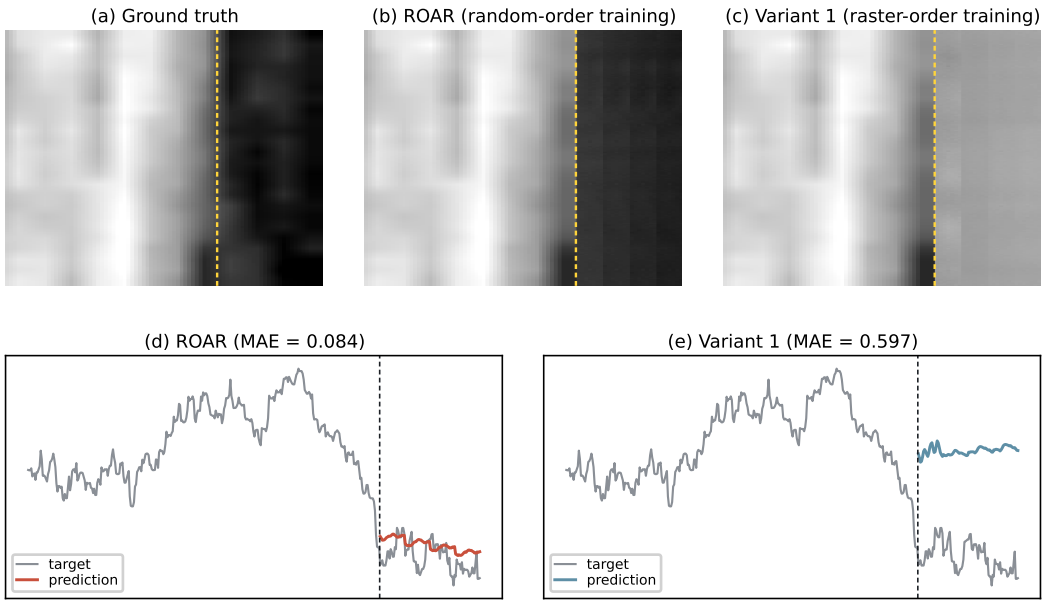

Figure 13: ROAR and Variant 1 on an Exchange test window. Images are cropped to the last eight days of history and the four future blocks (right of the dashed line). After the sharp drop at the end of the history, ROAR draws a dark future region that matches the ground truth, whereas Variant 1 draws a mid-grey region and forecasts a level well above the target.

G Hyperparameter Selection and Result Audit

Search protocol. The search varied tokenizer width/depth, predictor width/depth, learning rate, batch size, dropout, epoch budget, and curriculum parameters. A coarse capacity search was followed by local optimization and, where enabled, a curriculum search. Candidate checkpoints were selected by validation loss and then evaluated with the common four-horizon rolling protocol. The ranking criterion is average MSE across $H \in \{96, 192, 336, 720\}$; no horizon-specific checkpoint is selected.

Aggregation protocol. A result is eligible only when all requested horizon metrics are present and finite. Interrupted jobs, smoke tests, and incomplete exports are excluded. The aggregation script recomputes the four-horizon mean from horizon values. Seed groups require an exact configuration match and differ only in seed; neighboring hyperparameter candidates are retained as search evidence but are not treated as repeated trials.

Audit trail. Every retained run records the dataset file, dimensions, split interface, periodicity, context and target lengths, tokenizer checkpoints, predictor architecture, optimizer settings, curriculum settings, seed, selected checkpoint, and horizon-wise MSE/MAE. The downloaded audit set contains the raw training/evaluation output, JSON configuration, per-horizon CSV files, and summary exports. The ROAR-specific numerical tables in this appendix were recomputed from those exports; the comparative Average columns reproduce Table 1. The seven-dataset ablation table aggregates three seeds for each of four variants.

H Reproducibility Checklist

- **Data:** dataset name, dimensionality, native frequency, chronological split, and folding periodicity are reported in Table 3.
- **Model:** image size, six or seven history embeddings, four future latents, causal attention, order policy, normalization, frozen modules, and dataset-specific predictor sizes are specified above.

- **Optimization:** Adam learning rate, batch size, epoch budget, early-stopping patience, dropout, and the free-running warm-up are listed in Table 4; seed identities are reported in Section D.2.
- **Evaluation:** rollout chunk and context lengths, horizon set, target-free feedback, test-loader behavior, metric scale, scalar weighting, and averaging rule are explicitly defined.
- **Statistics:** horizon-level values, sample counts, and sample standard deviations are given in Table 5; the controlled ablation uses three seeds for every cell.

I LIMITATIONS

ROAR shares parameters across variables but does not explicitly model cross-variable interactions. Long horizons require recursive generation and can accumulate error; Exchange at $H = 720$ is the clearest example. The periodicity and predictor capacity are selected per dataset, which increases search cost and limits claims of a universal configuration. Deterministic inference uses one fixed serialization and does not provide calibrated predictive uncertainty. ROAR also trails the best baseline on Traffic, the benchmark with the most channels, where cross-variable modeling matters most.

The controlled component study covers seven datasets: five main-table datasets and the supplementary ETTh1 and ETTh2 datasets. Weather, Solar, and Traffic are not included. Exchange cannot identify a free-running effect because its selected full setting already disables that branch. Future work should evaluate cross-channel token interaction, probabilistic aggregation over valid orders, adaptive periodicity, irregular sampling, and fixed-budget multi-seed comparisons on every benchmark.