FACTS: A FACTORED STATE-SPACE FRAMEWORK FOR WORLD MODELLING

Li Nanbo[1]*, **Firas Laakom**[1], **Yucheng Xu**[2], **Wenyi Wang**[1], **Jürgen Schmidhuber**[1,3]
[1]Center of Excellence for Generative AI, KAUST, Saudi Arabia
[2]School of Informatics, University of Edinburgh, United Kingdom
[3]The Swiss AI Lab, IDSIA, USI & SUPSI, Switzerland

ABSTRACT

World modelling is essential for understanding and predicting the dynamics of complex systems by learning both spatial and temporal dependencies. However, current frameworks, such as Transformers and selective state-space models like Mambas, exhibit limitations in efficiently encoding spatial and temporal structures, particularly in scenarios requiring long-term high-dimensional sequence modelling. To address these issues, we propose a novel recurrent framework, the **FACT**ored **S**tate-space (**FACTS**) model, for spatial-temporal world modelling. The FACTS framework constructs a graph-structured memory with a routing mechanism that learns permutable memory representations, ensuring invariance to input permutations while adapting through selective state-space propagation. Furthermore, FACTS supports parallel computation of high-dimensional sequences. We empirically evaluate FACTS across diverse tasks, including multivariate time series forecasting and object-centric world modelling, demonstrating that it consistently outperforms or matches specialised state-of-the-art models, despite its general-purpose world modelling design.

1 INTRODUCTION

World modelling (Schmidhuber, 1990b; 2015; Ha & Schmidhuber, 2018) aims to create an internal representation of the environment for an AI system, enabling it to represent (Hafner et al., 2019; 2023), understand (Schrittwieser et al., 2020; Hafner et al., 2020), and predict (Ha & Schmidhuber, 2018; Micheli et al., 2022) the dynamics of complex environments. This capability is crucial for various domains, including autonomous systems, robotics, and financial forecasting, where accurate predictions depend on effectively capturing both spatial and temporal dependencies (Hafner et al., 2019; Ha & Schmidhuber, 2018). Consequently, spatial-temporal learning (Liu et al., 2024; Hochreiter & Schmidhuber, 1997; Wu et al., 2023a; Oreshkin et al., 2019) emerges as a key challenge in world modelling, as approaches must balance the complexities of modelling high-dimensional sequential data while maintaining robust long-term predictive power.

Despite significant advancements, current spatial-temporal learning frameworks, used in world modelling, based on Transformers (Vaswani et al., 2017; Schlag et al., 2021) and RNNs (Schmidhuber, 2015; Ha & Schmidhuber, 2018; Hafner et al., 2019; 2020) backbones, face limitations in fully capturing the complexities of high-dimensional spatial-temporal data. Transformer, though powerful (Chen et al., 2022; Robine et al., 2023; Micheli et al., 2022), are inefficient for long-term tasks due to their quadratic scaling and limited context windows (Zhang et al., 2022). On the other hand, RNNs provide a more structured approach to sequential data. However, their efficacy is hindered by the vanishing gradients (Hochreiter, 1991; Pascanu et al., 2013). The primary challenges in spatial-temporal learning arise from the high dimensionality of the data and the necessity to preserve long-term dependencies (Hochreiter et al., 2001; Tallec & Ollivier, 2018).

Recently, there has been a growing interest in Structured State-Space Models (SSMs) for world modelling (Gu & Dao, 2023; Hafner et al., 2023; Samsami et al., 2024) using latent state-space representations. These representations allow for the modelling of underlying dynamics, where latent

*Correspondence to nanbo.li@kaust.edu.sa

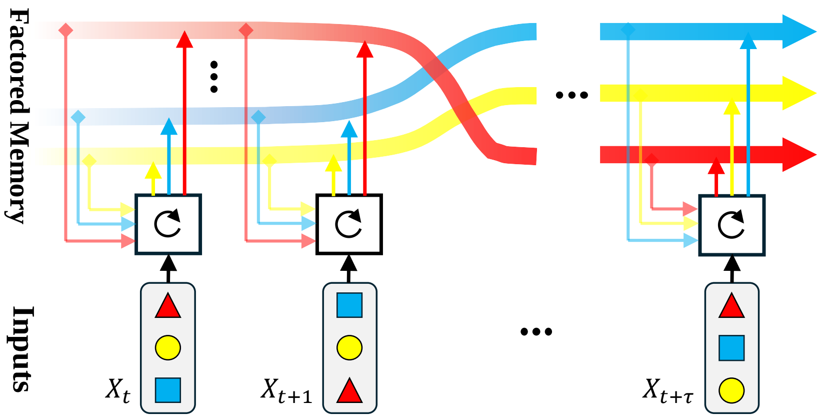


Figure 1: **Overview of FACTored State-space (FACTS) Architecture.** The FACTS framework constructs a factored state-space memory, allowing for flexible representations (e.g. graphs and sets). Sequential inputs (e.g. X_t) are processed through a selective memory-input interaction mechanism (denoted by the circular icon ↻), which determines how the inputs interact with and update factored memory. The different coloured pathways represent distinct latent factors, whose dynamics evolve over time based on these interactions. The design ensures that the memory update is permutation-invariant with respect to the input features, enabling FACTS to capture and track meaningful algorithmic regularities for accurate future predictions.

states evolve over time according to governing equations (Wang et al., 2024b). However, while SSM frameworks have demonstrated improved capacity for capturing temporal dynamics (Gu & Dao, 2023; Wang et al., 2024b; Baron et al., 2023), they often lack efficient mechanisms to handle high-dimensional spatial data. To address this limitation, recent SSMs often impose rigid structural constraints on their state spaces, such as diagonal (Gu & Dao, 2023; Gupta et al., 2022a;b) or block-diagonal structures (Dao & Gu, 2024), to capture invariant components throughout the sequence. This assumption, that specific dimensions of the state space correspond to consistent patterns over time, can be restrictive in world modelling scenarios where the relationship between state-space dimensions and input features evolves dynamically.

For example, in a dynamic system involving multiple agents (e.g. robots or sensors), where the positions of the agents change over time, to capture this dynamism, current SSMs require learning distinct representations for essentially identical scenarios at each time step as the agent locations change. This redundancy leads to an inefficient use of model capacity and data, ultimately limiting the model's ability to effectively capture the dynamics of the interactions between inputs and states. Therefore, there is a need for a consistent dynamic mapping between inputs and latent states to enhance spatial-temporal modelling capabilities and enable more efficient history compression (Schmidhuber, 1992a; 2003), which is essential for robust long-term prediction power. Another limitation of current SSM approaches is their inability to capture redundancy in the input space itself. In many cases, each agent's state (e.g., position, speed, direction) may contributes to the world's overall understanding, but the identity or order of the agents do not matter, i.e., only their interactions are crucial for making accurate predictions. In such instances, swapping agents should not alter the predictions and the world understanding. However, current SSM methods, typically based on linear transformations, fail to account for this and can perceive identical scenarios as different based on input order, hindering its ability to capture regularities and making them unsuitable for sequential modelling in various applications.

To address these challenges, in this paper, we propose the FACTS model, a novel recurrent framework for spatial-temporal world modelling. The FACTS model conceptualise the input as a set of nodes and introduces a permutable memory that can incorporate complex structures. Through selective memory-input routing, input features are dynamically assigned to distinct state-space factors, i.e., explanatory latent representations, that capture the underlying dynamics of the system. This formulation ensures input permutation invariance in the state-space memory, allowing the model to learn consistent factor representations over time, even when the spatial or temporal relationships between input features and factors change as illustrated in Figure 1. Additionally, by treating inputs as a set of nodes, FACTS: i) can incorporate a wider range of input structures, e.g., images or graphs or sets. ii) maintains consistent representations of inputs (e.g., agents), regardless of their order. This allows the model to capture regularities in both spatial/local modalities and temporal dependencies, enhancing its memory efficiency and long-term prediction capabilities through more efficient history compression (Schmidhuber, 1992a; 2003). To validate our proposed world modelling approach, we conduct an extensive empirical analysis across multiple tasks such as multivariate time series forecasting, and object-centric world modelling demonstrating that FACTS consistently matches or exceeds the performance of specialised state-of-the-art models. This confirms its robustness and versatility in addressing complex high-dimensional sequential tasks.

To sum up, our main contributions are as follows:

- We introduce **FACT**ored **S**tate-space (FACTS), a novel recurrent framework that incorporates a permutable memory structure, enabling flexible and efficient modelling of complex spatial-temporal dependencies.
- FACTS dynamically assigns input features to distinct latent state-space factors, ensuring effective history compression and enhancing long-term prediction power.
- We formally and empirically show that FACTS achieves consistent factor representations over time, regardless of changes in the spatial or temporal order of input features, providing robustness in dynamically evolving environments.
- We validate the robustness and predictive power of FACTS through extensive world modeling experiments, demonstrating its superior or competitive performance across multivariate time-series forecasting and object-centric world modeling tasks.

2 Preliminaries

Structured-state space Models (SSMs) have their roots in the classic Kalman filter (Kalman, 1960), where they process a m-dimensional input signal $\boldsymbol{x}(t) \in \mathbb{R}^m$ into a d-dimensional latent state $\boldsymbol{z}(t) \in \mathbb{R}^d$, which is then projected onto an output signal $\boldsymbol{y}(t) \in \mathbb{R}^n$. The general form of an SSM is expressed as follows:

$$\dot{\boldsymbol{z}}(t) = \boldsymbol{A}(t)\boldsymbol{z}(t) + \boldsymbol{B}(t)\boldsymbol{x}(t) \tag{1}$$

$$\boldsymbol{y}(t) = \boldsymbol{C}(t)\boldsymbol{z}(t) + \boldsymbol{D}(t)\boldsymbol{x}(t), \tag{2}$$

where $\dot{\boldsymbol{z}}(t) = \frac{d}{dt}\boldsymbol{z}(t)$ indicates the time derivative of the state. The matrices $\boldsymbol{A}(t) \in \mathbb{R}^{d\times d}$, $\boldsymbol{B}(t) \in \mathbb{R}^{d\times m}$, $\boldsymbol{C}(t) \in \mathbb{R}^{n\times d}$, and $\boldsymbol{D}(t) \in \mathbb{R}^{n\times m}$ present the state, input, output, and feed-forward matrices, respectively. In systems without direct feedthrough, $\boldsymbol{D}(t)$ becomes a zero matrix. Furthermore, since the original system operates in a continuous domain, discretisation is often used (Wang et al., 2024b; Smith et al., 2023), resulting in the general discrete-time formulation of SSM:

$$\boldsymbol{z}_t = \overline{\boldsymbol{A}}_t\boldsymbol{z}_{t-1} + \overline{\boldsymbol{B}}_t\boldsymbol{x}_t \tag{3}$$

$$\boldsymbol{y}_t = \boldsymbol{C}_t\boldsymbol{z}_t \tag{4}$$

with $\overline{\boldsymbol{A}}_t$, $\overline{\boldsymbol{B}}_t$, and $\boldsymbol{C}_t$ govern the dynamics driven by the input sequence $\boldsymbol{x}_{\leq t}$, with different constructions (Wang et al., 2024b; Gu & Dao, 2023; Dao & Gu, 2024) influencing the expressiveness and efficiency of the model. If we denote the state vector with $\boldsymbol{h}$, we can see that equation 3-equation 4 form is equivalent to the RNN dynamics. Hence, similarly to RNNs, the system, in equation 3-4, is inherently sequential, which inhibits parallel processing.

Parallelisation and the selective mechanism As shown in Blelloch (1990); Smith et al. (2023) if $\bar{\boldsymbol{B}}_t$ constructed independently of $\boldsymbol{z}_{t-1}$, the linear recurrence in equation 3 can be expanded as equation 5:

$$\boldsymbol{z}_t = \sum_{s=0}^{t} \bar{\boldsymbol{A}}^{\times}_{t:s}\bar{\boldsymbol{B}}_s\boldsymbol{x}_s, \tag{5}$$

where $\bar{\boldsymbol{A}}^{\times}_{t:s} \coloneqq \bar{\boldsymbol{A}}_{t+1}...\bar{\boldsymbol{A}}_{s+2}\bar{\boldsymbol{A}}_{s+1}$; $\bar{\boldsymbol{A}}_{t+1} = \boldsymbol{I}$; $\bar{\boldsymbol{B}}_0\boldsymbol{x}_0 = \boldsymbol{z}_0$ with respect to some initialisation. This expansion not only allows for parallel computation of the linear terms, but also reveals the direct connection established between distant inputs/observations along the sequential dimension, e.g. $\boldsymbol{x}_0$

and $\boldsymbol{x}_t$ with $t \gg 0$, thereby facilitating the capture of long-term dependencies. Furthermore, integrating a selective mechanism (Gu & Dao, 2023) by constructing $\bar{\boldsymbol{A}}_t$ and $\bar{\boldsymbol{B}}_t$ as functions of each input leads to the following formulation:

$$\boldsymbol{z}_t = \bar{\boldsymbol{A}}(\boldsymbol{x}_t)\boldsymbol{z}_{t-1} + \bar{\boldsymbol{B}}(\boldsymbol{x}_t)\boldsymbol{x}_t \tag{6}$$

$$= \sum_{s=0}^{t} \bar{\boldsymbol{A}}^{\times}(\boldsymbol{x}_{t:s})\bar{\boldsymbol{B}}(\boldsymbol{x}_s)\boldsymbol{x}_s \tag{7}$$

$$\boldsymbol{y}_t = \boldsymbol{C}(\boldsymbol{x}_t)\boldsymbol{z}_t \tag{8}$$

This formulation enables content-aware compression of the historical information, addressing the issue of memory decay in long-sequence modelling. Such selective mechanism is foundational to the effectiveness of modern state-space models (Gu & Dao, 2023; Dao & Gu, 2024). Additionally, as indicated in Eq. equation 6, SSMs can support parallel computation, since the term $\bar{\boldsymbol{B}}(\boldsymbol{x}_t)\boldsymbol{x}_t$ remains independent of the preceding state $\boldsymbol{z}_{t-1}$.

3 Proposed Framework: FACTS

In this section, we introduce our proposed framework: **FACTS** (**FACT**ored **S**tate-space) model, a novel class of recurrent neural networks designed with a structured state-space memory. FACTS is characterised by two key features: **permutable state-space memory**, which allows for flexible representation of system dynamics with more complex structures and **invariant recurrence** with respect to permutations of the input features, ensuring consistent modelling of underlying factors in the world across different time steps.

One key intuition behind the "permutable state-space memory" in FACTS is the principle of *history compression* (Schmidhuber, 1992a; 2003), which emphasises the need to eliminate redundant information in sequence modelling while uncovering algorithmic regularities. This principle is essential for effective long-sequence modelling with high-dimensional data, as it improves generalisation by reducing the accumulation of unnecessary information. Existing SSMs often address this challenge by imposing fixed structural constraints on their state spaces, such as diagonal or block-diagonal structures, to capture invariant components that persist throughout the sequence (Gu & Dao, 2023; Gupta et al., 2022a;b; Dao & Gu, 2024). However, these fixed structural priors assume that specific dimensions of the state space correspond to consistent and specific factors over time. This assumption can be limiting in world-modeling scenarios where the relationship between state-space dimensions and input features evolves dynamically. For instance, in video sequence modeling, factors may correspond to moving objects, and the spatial location of these objects (i.e., pixel positions) changes from frame to frame. In such cases, the model needs to adapt to these changes, but current SSMs formulations struggle to maintain consistent factor representations due to their rigid structural constraints, i.e., in equation 7 the matrices $\bar{\boldsymbol{B}}(\boldsymbol{x}_t)$ must not only select relevant information for modelling sequence dynamics but also account for the changing relative orders between subspaces of $\boldsymbol{z}_{t-1}$ and $\boldsymbol{x}_t$, which can evolve over time. This introduces additional complexity, leading to noise and redundancy that can hinder effective history compression.

3.1 FACTS formulation

The FACTS framework is formalised as a class of structured state-space model, which can capture the dynamic interactions between the latent factors and the input features. To facilitate this dynamic *factorisation*, i.e., the process of identifying and disentangling meaningful factors from the input data over time, at each time step t, we conceptualise the hidden state Z_t as a graph and hence, the state-space memory is represented as a set of nodes that correspond to the latent factors. The input features X_t are also treated as another set of nodes. Formally, let

$$Z_t = \{\boldsymbol{z}_t^1, \boldsymbol{z}_t^2, \ldots, \boldsymbol{z}_t^k\} \qquad X_t = \{\boldsymbol{x}_t^1, \boldsymbol{x}_t^2, \ldots, \boldsymbol{x}_t^m\} \tag{9}$$

where Z_t denote the set of k latent factors at time step t and m is the number of input features. By formulating both sets as nodes, FACTS is inherently invariant to permutations of both input features and factors. Then, to efficiently learn the optimal connections between these two sets, we propose a graph-based routing mechanism that can effectively match input features with the corresponding factors .i.e., learn edges between nodes in Z_t and X_t, reflecting the strength of each correspondence between each input feature and each latent factor.

Dynamic selective state-space updates In analogy with the standard SSM dynamics equation 3-equation 4, the evolution of the latent factors of FACTS is governed by the following modified state-space dynamics:

$$Z_t = \bar{\boldsymbol{A}}_t \odot Z_{t-1} + \bar{\boldsymbol{B}}_t \odot \boldsymbol{U}_t \tag{10}$$

$$\boldsymbol{y}_t = Dec(\boldsymbol{C}_t \odot Z_t) \tag{11}$$

Here, Z_t represents the state-space memory at time t, which stores the latent factors. The terms $\bar{\boldsymbol{A}}_t$, $\bar{\boldsymbol{B}}_t$, and $\bar{\boldsymbol{C}}_t$ are selective state-space model parameters responsible for controlling the information flow between the previous memory Z_{t-1} and the input features X_t. The symbol $\odot$ denotes element-wise multiplication, while Dec is a permutation-invariant decoder applied to the latent factors.

Compared to the standard SSM dynamics, i.e., equation 3-equation 4, we note two key differences: (i) FACTS relies on element-wise multiplication, instead of matrix multiplication, to conserve the invariance properties. (ii) x_t in equation 3 is replaced with $\boldsymbol{U}_t = (Z_{t-1}, X_t)$, which is a key element in FACTS that models the interactions between the memory Z_{t-1} and the input features X_t.

The attention-based router: Before diving into the details of the different parts of equation 10 and equation 11, we first introduce the routing mechanism used in this work. To maintain the recurrent permutability of equation 10, the routing mechanism between memory and inputs must dynamically assign input features to consistent factors. This can be done using an attention-based routing mechanism defined as follows:

$$Z_{t-1} \circlearrowright_{\phi,\psi,\varphi} X_t = \operatorname{softmax}\left(\frac{\phi(Z_{t-1})\psi^T(X_t)}{\sqrt{d}}\right)\varphi(X_t) \tag{12}$$

where the operator $\circlearrowright$ learns the relationships between the memory Z_{t-1} and input features X_t, dynamically determining which features correspond to which latent factors. The functions ϕ, ψ, and φ represent the query, key, and value mappings, respectively, and are applied row-wise to the memory and input features.

Factorisation process The term $\boldsymbol{U}_t = \boldsymbol{U}(Z_{t-1}, X_t)$ in equation 10 is crucial for capturing the interactions between the memory and the input features. Note that in prior works (Gu & Dao, 2023; Dao & Gu, 2024) $\boldsymbol{U}_t$ is typically constructed as function of the current input X_t only. In this paper, we argue that, similar to the gating in RNNs vs LSTMs (Hochreiter & Schmidhuber, 1997), it is more effective use both X_t and Z_{t-1} to conserve long term dependencies. This interaction plays a key role in factorisation, which refers to the process of binding the input features to specific memory items, effectively uncovering the underlying factors. In the FACTS framework, the memory at the previous time step Z_{t-1} serves as the prior over the latent factors, and $\boldsymbol{U}_t$ is the factor momentum that guides the evolution of these factors across time. This factor momentum is computed as:

$$\boldsymbol{U}_t = Z_{t-1} \circlearrowright_{\phi_U,\psi_U,\varphi_U} X_t \tag{13}$$

where ϕ_U, ψ_U, and φ_U are its corresponding query, key, and value mappings.

Selectivity through memory-input routing The selective state-space model parameters $\bar{\boldsymbol{A}}_t$, $\bar{\boldsymbol{B}}_t$, and $\bar{\boldsymbol{C}}_t$ are constructed through interactions between the memory and the input features, ensuring that both the memory and the inputs jointly decide which information should be retained or updated. These parameters are computed as follows:

$$\Delta_t = Z_{t-1} \circlearrowright_{\phi_\Delta,\psi_\Delta,\varphi_\Delta} X_t \qquad \bar{\boldsymbol{A}}_t = \exp(\alpha\Delta_t) \tag{14}$$

$$\bar{\boldsymbol{B}}_t = \Delta_t \odot (Z_{t-1} \circlearrowright_{\phi_B,\psi_B,\varphi_B} X_t) \qquad \bar{\boldsymbol{C}}_t = Z_{t-1} \circlearrowright_{\phi_C,\psi_C,\varphi_C} X_t \tag{15}$$

Here, Δ_t is a step size introduced for discretisation, and the functions ϕ_Δ, ψ_Δ, and φ_Δ (as well as their counterparts for $\bar{\boldsymbol{B}}_t$ and $\bar{\boldsymbol{C}}_t$) are responsible for mapping the memory and inputs to their respective selective parameters. The exponential function $\exp$ ensures that the selective parameters are non-negative, while α is a trainable scalar controlling the influence of Δ_t. By employing this selective mechanism, FACTS is capable of compressing long sequences in its state-space memory while maintaining the key properties of latent permutation equivariance and row-wise permutation invariance. Hence, FACTS can efficiently capture meaningful factors, e.g., objects in video frames or independent sources in a signal, even as their relationships with input features change over time.

Linearisation Although the framework presented so far in equation 10 has a permutable state-space memory and equivariant through the memory-input routing, which we formally show in Section 3.2, the routing between Z_{t-1} and X_t introduces dependencies of $\bar{\boldsymbol{B}}(Z_{t-1}, X_t)$ and $\boldsymbol{U}(Z_{t-1}, X_t)$ on Z_{t-1}. This results in a non-linear recurrence between Z_{t-1} and Z_t in equation 10, which limits parallelisation and hinders training efficiency. To overcome this, we substitute Z_{t-1} with Z_0, i.e. the initial memory or state, within the information routing processes. This leads to the final formulation of FACTS:

$$\begin{aligned} Z_t &= \mathbf{FACTS}(Z_{t-1}, Z_0, X_t) && (16) \\ &= \bar{\boldsymbol{A}}(Z_0, X_t) \odot Z_{t-1} + \bar{\boldsymbol{B}}(Z_0, X_t) \odot \boldsymbol{U}(Z_0, X_t) && (17) \\ &= \bar{\boldsymbol{A}}(Z_0, X_t) \odot \mathbf{FACTS}(Z_{t-2}, Z_0, X_{t-1}) + \bar{\boldsymbol{B}}(Z_0, X_t) \odot \boldsymbol{U}(Z_0, X_t) && (18) \\ &= \sum_{s=0}^{t} \bar{\boldsymbol{A}}^{\times}(Z_0, X_{t:s}) \odot \bar{\boldsymbol{B}}(Z_0, X_s) \odot \boldsymbol{U}(Z_0, X_s), && (19) \\ &= \mathbf{FACTS}(Z_0, X_{1:t}) && (20) \end{aligned}$$

where $\bar{\boldsymbol{A}}^{\times}(Z_0, X_{t:s}) = \bar{\boldsymbol{A}}_{t+1}...\bar{\boldsymbol{A}}_{s+2} \odot \bar{\boldsymbol{A}}_{s+1}$; the initial state Z_0 can either be provided a priori or sampled from unbiased distributions. This formulation in equation 17 linearise the recurrence in equation 10 by breaking the non-linear dependency between Z_t and Z_{t-1}. That is, as shown in equation 20, the inputs interact only with the initial memory, enabling fast computation of Z_t using equation 19, i.e., without recurrence. This significantly improves computational efficiency.

3.2 Theoretical Analysis of FACTS

Here, we formally proof the permutation equivariance and invariance properties of FACTS. We first formally define the two fundamental properties, namely left permutation equivariant (L.P.E.) and right permutation invariant (R.P.I.) in Definitions 1 and 2, respectively.

Definition 1. Let $f : \mathbb{R}^{n_1 \times n_2} \times \mathbb{R}^{t \times n_3 \times n_4} \to \mathbb{R}^{n_1 \times n_5}$ be a bivariate function with $n_1, n_2, n_3, n_4, n_5, t \in \mathbb{N}$. f is permutation equivariant (L.P.E.) if for all $\sigma \in S_{n_1}, \boldsymbol{M}_1 \in \mathbb{R}^{n_1 \times n_2}$, and $\boldsymbol{M}_2 \in \mathbb{R}^{t \times n_3 \times n_4}$,

$$f(\sigma \boldsymbol{M}_1, \boldsymbol{M}_2) = \sigma f(\boldsymbol{M}_1, \boldsymbol{M}_2),$$

where S_k denotes the set of permutation matrices of size $\mathbb{R}^{k \times k}$.

Definition 2. Let $\mathbb{R}^{n_1 \times n_2} \times \mathbb{R}^{t \times n_3 \times n_4} \to \mathbb{R}^{n_1 \times n_5}$ be a bivariate function with $n_1, n_2, n_3, n_4, n_5, t \in \mathbb{N}$. f is right permutation invariant (R.P.I.) if for all $\sigma_1, \sigma_2, \ldots, \sigma_t \in S_{n_3}, \boldsymbol{M}_1 \in \mathbb{R}^{n_1 \times n_2}$, and $\boldsymbol{M}_2^1, \boldsymbol{M}_2^2, \ldots, \boldsymbol{M}_2^t \in \mathbb{R}^{n_3 \times n_4}$,

$$f(\boldsymbol{M}_1, [\sigma_1 \boldsymbol{M}_2^1, \sigma_2 \boldsymbol{M}_2^2, \ldots, \sigma_t \boldsymbol{M}_2^t]) = f(\boldsymbol{M}_1, [\boldsymbol{M}_2^1, \boldsymbol{M}_2^2, \ldots, \boldsymbol{M}_2^t]).$$

These L.P.E. and R.P.I. properties, which formally describe the two fundamental aspects of FACTS: *permutable memory* and *permutation-invariant recurrence (w.r.t. the features)* — with memory Z_{t-1} and X_t serving as the left and right arguments of FACTS. They are thus essential not only for constructing the routing mechanism but also for the overall design of FACTS.

Using Definitions 1 and 2, that by taking memory Z_t and features X_t as the left and right arguments in FACTS (equation 20), we can show the following result:

Theorem 1. $\mathbf{FACTS}$ *as defined in equation 20 is L.P.E. and R.P.I.*

The proof of Theorem 1 is available in Appendix B. Theorem 1 proves our main claim that FACTS: i) is invariant to input features permutation. ii) learns permutable state-space memory. Furthermore, it is possible to extend our results in Theorem 1 to the more general case, where $\bar{\boldsymbol{A}}, \bar{\boldsymbol{B}}, \boldsymbol{U}$ are L.P.E. and R.P.I. functions of Z_{t-1} and X_t. The main result is presented in Theorem 2.

Theorem 2. *if $\bar{\boldsymbol{A}}, \bar{\boldsymbol{B}}, \boldsymbol{U}$ are L.P.E. and R.P.I. functions of Z_{t-1} and X_t, any dynamics governed by equation 10 is L.P.E. and R.P.I.*

The proof of Theorem 2 is available in Appendix B. Theorem 2 highlights the main condition on the variables $\bar{\boldsymbol{A}}, \bar{\boldsymbol{B}}, \boldsymbol{U}$ to ensure that the model is invariant to input features and has an equivariant memory. This can spark future research to develop SSM models based on equation 10 that are efficient history compressors and are suitable to dynamic world modelling scenarios.

4 Experiments

We design experiments to evaluate the effectiveness of FACTS in world modelling. We frame world modelling as a prediction task, where the model must predict future events in complex environments based on observed history, and evaluate a model's performance by its prediction accuracy. We conduct experiments on two environments: the multivariate time series (MTS) benchmark for forecasting and the CLEVRER multi-object video dataset for video prediction.

4.1 Long term forecasting

In many real-world applications, such as climate prediction, traffic flow management, or autonomous systems, predicting future states over long horizons is crucial for effective decision-making. World modeling, in these domains, often involves high-dimensional multivariate inputs—such as interacting agents, variables, or environmental factors—requiring the system to account for their complex dependencies and interactions across time. Long-term forecasting in this context is challenging as it demands accurate representation of temporal dynamics over extended periods. Additionally, input features may lack a predefined order, or this order could change dynamically. For example, a system may receive data from sensors (e.g., temperature and pressure) without knowing which is which during testing. The world modeler must provide reliable long-term predictions even if the input order changes unexpectedly, and generalize to unseen configurations, without learning every possible permutation of input features during training.

Benchmark We use the open-source Time Series Library (TSLib)[1], a widely-used benchmark for training and evaluating time-series models. TSLib provides standardized settings and a leaderboard of top-performing models, ensuring fair and consistent comparisons. Our focus is on long-term multivariate time-series forecasting (MSTF) tasks, using 9 diverse real-world datasets: 4 ETT datasets, Electricity, Weather, Exchange, Traffic, and Solar-Energy. Our approach FACTS is compared against 8 baseline models, including state-of-the-art MSTF approaches that top the TSLib leaderboard (Wang et al., 2024c; Liu et al., 2024; Wu et al., 2023a; Nie et al., 2023; Zeng et al., 2023; Zhang & Yan, 2023; Zhou et al., 2022; Wu et al., 2021). Following the setup of iTransformer and S-Mamba, we fix the input sequence length to 96 and evaluate the models on prediction lengths of 96, 192, 336, 720. The datasets and experimental protocols are widely used in MSTF literature, providing a robust evaluation framework. Prediction accuracy is measured using mean-squared error (MSE) and mean-absolute error (MAE) as the primary metrics.

4.1.1 Forecasting with predefined order (Scenario 1)

We use the exact same setup to Wang et al. (2024c); Liu et al. (2024); Wu et al. (2023a), with the exception of the pre- and post-processing modules (referred to as the "embedders" and "projectors" in TSLib). In our implementation, we replace these with set functions to accommodate the output structure of FACTS (c.f. Appendix C for more details). Note that in the standard setup of TSLib, the arrangement of the input features in the test is not changed and is the identical to the arrangement to the one seen during the training. The average results over the different prediction windows of our proposed approach along with all competing methods are presented in Table 1 and the full results are available in Table 6 in Appendix D.

Table 1 highlights the strong performance of FACTS, which achieves competitive results in both metrics, compared to the competing state-of-the-art specialized MSTF models. For instance, in terms of MAE, FACTS achieves the highest scores on 6 out of 9 datasets and is always in the top 2 in 7 out of 9 them. Even where it is not the top performer, FACTS remains highly competitive (3rd place) as seen in Traffic and Solar-Energy underscoring its robustness and ability to adapt to different scenarios. FACTS ability to capture long-term dependencies efficiently can be attributed to its structured state-space memory, which it encourages better learning statistical independence factors whose interactions can explain the spatial-temporal correlations of the multivariate observations.

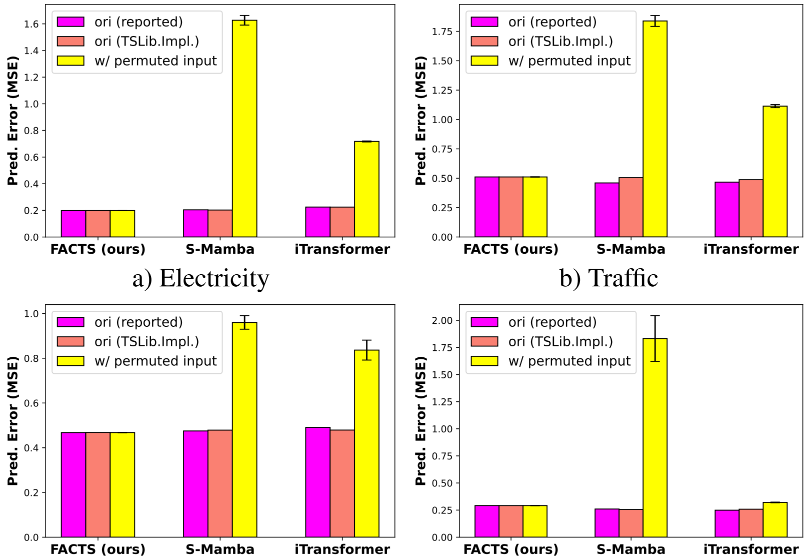
4.1.2 Forecasting with unknown Order (Scenario 2)

To evaluate robustness under dynamic scenarios, we randomly permute the input features during the test phase to simulate environments where the arrangement of agents or entities (e.g., robots, sensors) changes unpredictably. This mirrors real-world scenarios where input configurations vary, challenging world models to adapt to unseen input orderings. We focus on top pretrained models

[1] Time Series Library benchmark: `https://github.com/thuml/Time-Series-Library.git`

		ETTm1	ETTm2	ETTh1	ETTh2	Electricity	Exchange	Traffic	Weather	Solar-Energy
Autoformer (2021)	**MSE**	0.588	0.327	0.496	0.450	0.227	0.613	0.628	0.338	0.885
	MAE	0.617	0.371	0.487	0.459	0.338	0.539	0.379	0.382	0.711
FEDformer (2022)	**MSE**	0.448	0.305	**0.440**	0.437	0.214	0.519	0.610	0.309	0.291
	MAE	0.452	0.349	0.460	0.449	0.327	0.429	0.376	0.360	0.381
TimesNet (2023a)	**MSE**	0.400	0.291	0.458	0.414	0.192	0.416	0.620	0.259	0.301
	MAE	0.406	0.333	0.450	0.427	0.295	0.443	0.336	0.287	0.319
PatchTST (2023)	**MSE**	**0.387**	**0.281**	0.469	0.387	0.205	0.367	0.481	0.259	0.270
	MAE	0.400	0.326	0.454	0.407	0.290	0.404	0.304	0.281	0.307
DLinear (2023)	**MSE**	0.403	0.350	0.456	0.559	0.212	**0.354**	0.625	0.265	0.330
	MAE	0.407	0.401	0.452	0.515	0.300	0.414	0.383	0.317	0.401
Crossformer (2023)	**MSE**	0.513	0.757	0.529	0.942	0.244	0.940	0.550	0.259	0.641
	MAE	0.496	0.610	0.522	0.684	0.334	0.707	0.304	0.315	0.639
iTransformer (2024)	**MSE**	0.407	0.288	0.454	0.383	0.178	0.360	0.428	0.258	**0.233**
	MAE	0.410	0.332	0.447	0.407	0.270	0.403	0.282	0.278	**0.262**
S-Mamba (2024c)	**MSE**	0.398	0.288	0.455	0.381	0.170	0.367	**0.414**	0.251	0.240
	MAE	0.405	0.332	0.450	0.405	0.265	0.408	**0.276**	**0.276**	0.273
FACTS (Ours)	**MSE**	0.393	**0.281**	0.441	**0.376**	**0.168**	0.355	0.470	**0.250**	0.256
	MAE	**0.399**	**0.325**	**0.428**	**0.399**	**0.264**	**0.398**	0.298	0.277	0.274

Table 1: average MSE and MAE errors of the different approaches on the multivariate time series forecasting tasks. For each metric and each dataset, the top performance and the second best are highlighted in **red** and blue, respectively.



a) Electricity

b) Traffic

c) ETTm1

d) Solar Energy

Figure 2: Model Robustness to Input Permutations on 4 datasets from MTSF. Magenta bars represent original performance, salmon bars show performance using our/TSLib implementation, and yellow bars represent results under input permutation. Results are averaged over five random seeds, with error bars showing ±2× standard deviation.

from the first scenario (Table 1), i.e., FACTS, iTransformer, and S-Mamba. For the datasets, we use the challenging ones from the first scenario and permute the feature embeddings five times during testing, reporting the average performance and two standard deviations.

The main results are presented in Figure 2. While other models, iTransformer and S-Mamba, experience significant degradation in performance when the input features order is shuffled, we note that FACTS consistently maintains its prediction performance across the different tasks. These findings corroborate the theoretical results of Section 3.2. For example, on the Traffic dataset, the MSE errors of S-mamba, which is the model with the top performance in the standard setting (Table 1), increases more than threefold, and the error for iTransformer doubles. In contrast, FACTS, leveraging its selective memory routing which consistently assigns input features to the latent factors, preserves low error rates despite the permutation. This highlights FACTS' ability to handle dynamic and unordered environments. This adaptability further emphasizes the generalization strength of FACTS, particularly in world modeling scenarios where input orders may be inconsistent or unknown.

Method	LPIPS ↓
PredRNN (2017)	0.17
VQFormer (2019)	0.18
G-SWM (2020)	0.16
SlotFormer (2023b)	0.11
SAVi-dyn (2023)	0.19
FACTS (Ours)	**0.09**

Method	ARI ↑	FG-ARI ↑	FG-mIoU ↑
G-SWM (2020)	57.14	49.61	24.44
SlotFormer (2023b)	**63.45**	63.00	29.81
SAVi-dyn (2023)	8.64	**64.32**	18.25
FACTS (Ours)	58.25	62.34	**48.11**

Table 2: Quantitative results of object dynamics prediction. **Left**: visual quality of the predicted future frames, measured by LPIPS (lower is better). **Right**: segmentation of predicted future frames using ARI, FG-ARI, and FG-mIoU (higher is better).

4.2 Object-centric world modelling

Visual object-centric representation learning (OCRL) (Burgess et al., 2019; Greff et al., 2019; Locatello et al., 2020; Nanbo et al., 2020) tackles the challenge of binding visual information to consistent factors, even as object features dynamically permute with movement across pixels in videos (Kipf et al., 2023). This aligns with FACTS' objective of identifying regularities in dynamic environments for history compression and future-event prediction, making OCRL an ideal evaluation benchmark. To evaluate how FACTS 1) leverages object information for future predictions and 2) aligns its discovered factors with objects, we conduct two OCRL experiments, *slot dynamics prediction* and *unsupervised object discovery*, set on the simulated CLEVRER multi-object videos (Yi et al., 2020).

Slot dynamics prediction The task involves having a world model capture object-centric dynamics in latent space: given the latent object representations of observed events ("burn-in"), the model predicts the future latent codes of the objects ("roll-out"). We conducted this experiment following the setup of our major baseline, SlotFormer (Wu et al., 2023b). We evaluate the performance of the model by assessing 1) the visual quality of the predicted future frames and 2) the precision of future segmentation map rendered from the predicted latents. We quantify visual quality using the LPIPS metric, which provides stronger alignment with human perception than other commonly used metics such as PSNR and SSIM (Wu et al., 2023b; Sara et al., 2019), and segmentation accuracy using the commonly used Mean Intersection over Union (mIoU) and Adjusted Rand Index (ARI), with and/or without the foreground focus.
Results The results presented in Table 2 highlight the strengths of the FACTS model in terms of both visual quality and segmentation accuracy for object dynamics prediction on the CLEVRER dataset. FACTS achieves the lowest LPIPS score of 0.09, indicating superior visual quality in the predicted frames. Additionally, it demonstrates competitive performance in segmentation accuracy, attaining a leading FG-mIoU of 48.11, which highlights its effectiveness in predicting object positions and interactions in future frames. We attribute these results to FACTS' selective history compression mechanism. In contrast to SlotFormer, which predicts the next state by attending to all past inputs—resulting in inefficiency and noisy predictions—FACTS effectively compresses and retains only the most relevant information in memory, thereby filtering out noise and yielding more accurate dynamics modelling and future predictions.

Unsupervised object discovery In contrast to the slot dynamics task, where object slots or factors are given as input, this experiment requires FACTS to autonomously discover relevant factors for future predictions in multi-object videos. This process enables us to understand the regularities that FACTS identifies as significant for forecasting future events. We utilise a CNN vision encoder to convert the input image into a feature set, from which FACTS learns the object factors. These factors are employed to predict future object slots in an autoregressive manner and are subsequently decoded back into video frames. Although we adopt the architecture of SAVi's vision encoder and slot decoder, all modules are trained end-to-end from scratch by jointly minimising the reconstruction MSE and future prediction MSE. This approach highlights a key distinction from models like SAVi, which primarily focus on object identification for reconstructing observations, whereas FACTS aims to discover the regularities or modularities that enhance future predictions. Consequently, FACTS and SAVi conceptualise "factors" differently.
Results We visualise the discovered factors by independently rendering each factor's dynamics back into video while keeping the others fixed. As shown in Figure 5 (in the Appendix), FACTS primar-

Pred.Len	MS-Conv2d Emb.	Conv2d Emb.	DFT Emb.	Avg±Std.Err.
96	0.143	0.144	0.147	0.145±0.002
192	0.158	0.159	0.161	0.159±0.001
336	0.171	0.170	0.171	0.171±0.000
720	0.198	0.228	0.236	0.219±0.015

Table 3: Ablation: MSTF Performance of FACTS vs. different embedders (MSE↓).

$k \backslash d$	1	128	512	Avg.±Std.Err.
1	0.349	0.330	0.328	0.336±0.007
3	0.349	0.337	0.326	0.337±0.007
5	0.349	0.331	0.349	0.343±0.006
7	0.349	0.331	0.330	0.337±0.006
9	0.349	0.332	0.352	0.344±0.006
Avg.±Std.Err.	0.349±0.000	0.332±0.001	0.336±0.006	—

Table 4: Ablation: MSTF Performance of FACTS vs. (#factors k, #dimensions d) (MSE↓).

ily identifies moving objects—those deemed "useful" for future predictions—while treating static objects as background. This behaviour makes direct comparisons with existing unsupervised object discovery models challenging, we thus show only qualitative results in Appendix D and our accompanying video demo.

4.3 Ablation Study

FACTS requires the use of a set encoder and the predefined selection of the number of factors in the state-space memory prior to training. Our ablation study aims to investigate the impact of different set encoders and the choice of the number of predefined factors on model performance.

Impact of different set encoders We conduct our experiments on the Electricity dataset, testing four prediction lengths: 96, 192, 336, and 720. Three different set encoders are evaluated, each employing different priors: a Discrete-Fourier Transform (DFT) decomposer, a trainable Conv2d embedder, and a multi-scale Conv2d embedder (inspired by Wu et al. (2023a)). Table 3 presents the comparison results, with further details of these embedders provided in Appendix C.2. The multi-scale periodic embedder consistently outperforms the others across all prediction lengths, while the DFT-based embedder shows declining performance as the prediction length increases. The standard error further indicates that longer forecasting horizons amplify the impact of encoder choice, making it a more critical factor in model accuracy. This highlights the importance of using an unbiased, learnable set encoder to improve generalisation.

Impact of different number of factors We investigate the effect of the preset number of factors (k) on FACTS' performance, using the ETTm1 dataset with a 96-prediction length setting. Previous work, such as Mamba (Gu & Dao, 2023) and xLSTM (Beck et al., 2024), shows that state-space memory size significantly impacts performance. In FACTS, memory size is determined by the number of factors and the dimension of each factor (d). To isolate the effect of the number of factors, we trained the model across various settings, gridded by different numbers of factors and factor dimensions. As shown in Table 4, FACTS achieves consistent performance and is indeed robust to the hyper-parameters.

5 Discussions & Conclusion

In this work, we introduced FACTS, a novel recurrent framework designed for spatial-temporal world modelling. FACTS is constructed permutable state-space memory, which offers the flexibility needed to capture complex dependencies across time and space. By employing selective memory-input routing, FACTS is able to dynamically assign input features to distinct latent factors, enabling more efficient history compression and long-term prediction accuracy. Furthermore, we formally showed that FACTS: i) is invariant to input features permutation. ii) learns permutable state-space memory, maintaining consistent factor representations regardless of changes in the input order. Furthermore, through comprehensive empirical evaluations, FACTS demonstrated superior performance on a variety of real-world datasets, consistently matching or outperforming specialized state-of-the-art models in diverse tasks. Notably, FACTS maintained its predictive powers even in challenging settings where the order of input features was shuffled, highlighting its robustness and adaptability. These results underscore the model's potential for a wide range of applications, particularly in world modeling scenarios where input configurations are variable or uncertain. For future work, we plan to extend FACTS to larger-scale experiments, exploring its scalability and potential in even more complex world modeling tasks.

ACKNOWLEDGEMENT

The research reported in this publication was supported by funding from King Abdullah University of Science and Technology (KAUST)-Center of Excellence for Generative AI, under award number 5940.

REFERENCES

Ethan Baron, Itamar Zimerman, and Lior Wolf. A 2-dimensional state space layer for spatial inductive bias. In *The Twelfth International Conference on Learning Representations*, 2023.

Maximilian Beck, Korbinian Pöppel, Markus Spanring, Andreas Auer, Oleksandra Prudnikova, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. xLSTM: Extended long short-term memory. *arXiv preprint arXiv:2405.04517*, 2024.

Guy E Blelloch. Prefix sums and their applications. 1990.

Christopher P Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. MONet: Unsupervised scene decomposition and representation. *arXiv preprint arXiv:1901.11390*, 2019.

Chang Chen, Yi-Fu Wu, Jaesik Yoon, and Sungjin Ahn. Transdreamer: Reinforcement learning with transformer world models. *arXiv preprint arXiv:2202.09481*, 2022.

Tianxiang Chen, Zhentao Tan, Tao Gong, Qi Chu, Yue Wu, Bin Liu, Jieping Ye, and Nenghai Yu. MiM-ISTD: Mamba-in-Mamba for efficient infrared small target detection. *arXiv preprint arXiv:2403.02148*, 2024.

Silvia Chiappa, Sébastien Racaniere, Daan Wierstra, and Shakir Mohamed. Recurrent environment simulators. *arXiv preprint arXiv:1704.02254*, 2017.

Kyunghyun Cho. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. *arXiv preprint arXiv:2405.21060*, 2024.

Marc Deisenroth and Carl E Rasmussen. PILCO: A model-based and data-efficient approach to policy search. In *International Conference on machine learning*, 2011.

Emily L Denton et al. Unsupervised learning of disentangled representations from video. *Advances in neural information processing systems*, 30, 2017.

Stefan Depeweg, José Miguel Hernández-Lobato, Finale Doshi-Velez, and Steffen Udluft. Learning and policy search in stochastic dynamical systems with Bayesian neural networks. *arXiv preprint arXiv:1605.07127*, 2016.

Chengbin Du, Yanxi Li, and Chang Xu. Understanding robustness of visual state space models for image classification. *arXiv preprint arXiv:2403.10935*, 2024a.

Yu Du, Xu Liu, and Yansong Chua. Spiking structured state space model for monaural speech enhancement. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 766–770. IEEE, 2024b.

Zhengcong Fei, Mingyuan Fan, Changqian Yu, and Junshi Huang. Scalable diffusion models with state space backbone. *arXiv preprint arXiv:2402.05608*, 2024.

Chelsea Finn, Xin Yu Tan, Yan Duan, Trevor Darrell, Sergey Levine, and Pieter Abbeel. Deep spatial autoencoders for visuomotor learning. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 512–519. IEEE, 2016.

Yarin Gal, Rowan McAllister, and Carl Edward Rasmussen. Improving PILCO with Bayesian neural network dynamics models. In *Data-efficient machine learning workshop, ICML*, volume 4, pp. 25, 2016.

Zoubin Ghahramani. Factorial learning and the EM algorithm. *Advances in neural information processing systems*, 7, 1994.

Riccardo Grazzi, Julien Siems, Simon Schrodi, Thomas Brox, and Frank Hutter. Is Mamba capable of in-context learning? *arXiv preprint arXiv:2402.03170*, 2024.

Klaus Greff, Sjoerd Van Steenkiste, and Jürgen Schmidhuber. Neural expectation maximization. In *Advances in Neural Information Processing Systems*, pp. 6691–6701, 2017.

Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 2424–2433, 2019.

Klaus Greff, Sjoerd Van Steenkiste, and Jürgen Schmidhuber. On the binding problem in artificial neural networks. *arXiv preprint arXiv:2012.05208*, 2020.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021.

Ankit Gupta, Albert Gu, and Jonathan Berant. Diagonal state spaces are as effective as structured state spaces. *Advances in Neural Information Processing Systems*, 35:22982–22994, 2022a.

Ankit Gupta, Harsh Mehta, and Jonathan Berant. Simplifying and understanding state space models with diagonal linear RNNs. *arXiv preprint arXiv:2212.00768*, 2022b.

David Ha and Jürgen Schmidhuber. World models. *Preprint arXiv:1803.10122 (variant at NeurIPS 2018)*, 2018.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pp. 2555–2565. PMLR, 2019.

Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering Atari with discrete world models. *arXiv preprint arXiv:2010.02193*, 2020.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Wei He, Kai Han, Yehui Tang, Chengcheng Wang, Yujie Yang, Tianyu Guo, and Yunhe Wang. DenseMamba: State space models with dense hidden connection for efficient large language models. *arXiv preprint arXiv:2403.00818*, 2024.

Daniel Hein, Stefan Depeweg, Michel Tokic, Steffen Udluft, Alexander Hentschel, Thomas A Runkler, and Volkmar Sterzing. A benchmark environment motivated by industrial control problems. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1–8. IEEE, 2017.

Sepp Hochreiter. Untersuchungen zu dynamischen neuronalen Netzen. *Diploma, Technische Universität München*, 91(1):31, 1991.

Sepp Hochreiter and Jürgen Schmidhuber. Bridging long time lags by weight guessing and "long short-term memory". *Spatiotemporal models in biological and artificial systems*, 37(65-72):11, 1996.

Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8): 1735–1780, 1997.

Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, and Jürgen Schmidhuber. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, 2001.

Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.

Tao Huang, Xiaohuan Pei, Shan You, Fei Wang, Chen Qian, and Chang Xu. LocalMamba: Visual state space model with windowed selective scan. *arXiv preprint arXiv:2403.09338*, 2024.

Md Mohaiminul Islam and Gedas Bertasius. Long movie clip classification with state-space video models. In *European Conference on Computer Vision*, pp. 87–104. Springer, 2022.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2901–2910, 2017.

Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.

Diederik P Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Thomas Kipf, Elise Van der Pol, and Max Welling. Contrastive learning of structured world models. In *International Conference on Learning Representations*, 2019.

Thomas Kipf, Gamaleldin Fathy Elsayed, Aravindh Mahendran, Austin Stone, Sara Sabour, Georg Heigold, Rico Jonschkowski, Alexey Dosovitskiy, and Klaus Greff. Conditional object-centric learning from video. In *International Conference on Learning Representations*, 2023.

Igor Kononenko. Bayesian neural networks. *Biological Cybernetics*, 61(5):361–370, 1989.

Zhixuan Lin, Yi-Fu Wu, Skand Peri, Bofeng Fu, Jindong Jiang, and Sungjin Ahn. Improving generative imagination in object-centric world models. In *International Conference on Machine Learning*, pp. 6140–6149. PMLR, 2020.

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 *International Conference on Learning Representations*, 2024.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *Advances in Neural Information Processing Systems*, 2020.

Xianping Ma, Xiaokang Zhang, and Man-On Pun. RS 3 Mamba: Visual state space model for remote sensing image semantic segmentation. *IEEE Geoscience and Remote Sensing Letters*, 2024.

David JC MacKay et al. Introduction to Gaussian processes. *NATO ASI series F computer and systems sciences*, 168:133–166, 1998.

Rowan McAllister and Carl Edward Rasmussen. Data-efficient reinforcement learning in continuous-state POMDPs. *arXiv preprint arXiv:1602.02523*, 2016.

Harsh Mehta, Ankit Gupta, Ashok Cutkosky, and Behnam Neyshabur. Long range language modeling via gated state spaces. *arXiv preprint arXiv:2206.13947*, 2022.

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models. *arXiv preprint arXiv:2209.00588*, 2022.

Paul Munro. A dual back-propagation scheme for scalar reward learning. In *Ninth Annual Conference of the Cognitive Science Society*, pp. 165–176. Hillsdale, NJ. Cognitive Science Society Lawrence Erlbaum, 1987.

Li Nanbo, Cian Eastwood, and Robert Fisher. Learning object-centric representations of multi-object scenes from multiple views. In *Advances in Neural Information Processing Systems*, 2020.

Derrick Nguyen and Bernard Widrow. The truck backer-upper: An example of self-learning in neural networks. In *Advanced neural computers*, pp. 11–19. Elsevier, 1990.

Eric Nguyen, Karan Goel, Albert Gu, Gordon Downs, Preey Shah, Tri Dao, Stephen Baccus, and Christopher Ré. S4ND: Modeling images and videos as multidimensional signals with state spaces. *Advances in neural information processing systems*, 35:2846–2861, 2022.

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

Junhyuk Oh, Xiaoxiao Guo, Honglak Lee, Richard L Lewis, and Satinder Singh. Action-conditional video prediction using deep networks in Atari games. *Advances in neural information processing systems*, 28, 2015.

Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. *arXiv preprint arXiv:1905.10437*, 2019.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. pp. III–1310, 2013.

Thomas Porter and Tom Duff. Compositing digital images. In *Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, pp. 253–259, 1984.

Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with VQ-VAE-2. In *Advances in neural information processing systems*, 2019.

Jan Robine, Marc Höftmann, Tobias Uelwer, and Stefan Harmeling. Transformer-based world models are happy with 100k interactions. *arXiv preprint arXiv:2303.07109*, 2023.

Tony Robinson and Frank Fallside. Dynamic reinforcement driven error propagation networks with application to game playing. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 11, 1989.

David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International journal of forecasting*, 36(3): 1181–1191, 2020.

Mohammad Reza Samsami, Artem Zholus, Janarthanan Rajendran, and Sarath Chandar. Mastering memory tasks with world models. *arXiv preprint arXiv:2403.04253*, 2024.

Umme Sara, Morium Akter, and Mohammad Shorif Uddin. Image quality assessment through FSIM, SSIM, MSE and PSNR—a comparative study. *Journal of Computer and Communications*, 7(3):8–18, 2019.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear transformers are secretly fast weight programmers. In *International Conference on Machine Learning*, pp. 9355–9366. PMLR, 2021.

Karl Schmeckpeper, Annie Xie, Oleh Rybkin, Stephen Tian, Kostas Daniilidis, Sergey Levine, and Chelsea Finn. Learning predictive models from observation and interaction. In *European Conference on Computer Vision*, pp. 708–725. Springer, 2020.

Jürgen Schmidhuber. An on-line algorithm for dynamic reinforcement learning and planning in reactive environments. In *1990 IJCNN international joint conference on neural networks*, pp. 253–258. IEEE, 1990a.

Jürgen Schmidhuber. Making the world differentiable: On using fully recurrent self-supervised neural networks for dynamic reinforcement learning and planning in non-stationary environments. *Institut für Informatik, Technische Universität München. Technical Report FKI-126*, 90, 1990b.

Jürgen Schmidhuber. Reinforcement learning in Markovian and non-Markovian environments. In D. S. Lippman, J. E. Moody, and D. S. Touretzky (eds.), *Advances in Neural Information Processing Systems 3 (NIPS 3)*, pp. 500–506. Morgan Kaufmann, 1991a.

Jürgen Schmidhuber. A possibility for implementing curiosity and boredom in model-building neural controllers. pp. 222–227, 1991b.

Jürgen Schmidhuber. Learning complex, extended sequences using the principle of history compression. *Neural computation*, 4(2):234–242, 1992a.

Jürgen Schmidhuber. Learning factorial codes by predictability minimization. *Neural computation*, 4(6):863–879, 1992b.

Jürgen Schmidhuber. Exploring the predictable. In *Advances in evolutionary computing: theory and applications*, pp. 579–612. Springer, 2003.

Jürgen Schmidhuber. On learning to think: Algorithmic information theory for novel combinations of reinforcement learning controllers and recurrent neural world models. *arXiv preprint arXiv:1511.09249*, 2015.

Jürgen Schmidhuber, Daan Wierstra, Matteo Gagliolo, and Faustino Gomez. Training recurrent networks by EVOLINO. *Neural computation*, 19(3):757–779, 2007.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering Atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

Yuan Shi, Bin Xia, Xiaoyu Jin, Xing Wang, Tianyu Zhao, Xin Xia, Xuefeng Xiao, and Wenming Yang. VmambaIR: Visual state space model for image restoration. *arXiv preprint arXiv:2403.11423*, 2024.

David Silver, Hado Hasselt, Matteo Hessel, Tom Schaul, Arthur Guez, Tim Harley, Gabriel Dulac-Arnold, David Reichert, Neil Rabinowitz, Andre Barreto, et al. The Predictron: End-to-end learning and planning. In *International Conference on Machine Learning*, pp. 3191–3199. PMLR, 2017.

Jimmy TH Smith, Andrew Warrington, and Scott W Linderman. Simplified state space layers for sequence modeling. *International Conference on Learning Representations*, 2023.

Rupesh K Srivastava, Klaus Greff, and Jürgen Schmidhuber. Training very deep networks. *Advances in neural information processing systems*, 28, 2015.

Aleksandar Stanić, Yujin Tang, David Ha, and Jürgen Schmidhuber. Learning to generalize with object-centric agents in the open world survival game crafter. *IEEE Transactions on Games*, 2023.

Corentin Tallec and Yann Ollivier. Can recurrent neural networks warp time? *arXiv preprint arXiv:1804.11188*, 2018.

Michael E Tipping and Christopher M Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 61(3):611–622, 1999.

Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

Jue Wang, Wentao Zhu, Pichao Wang, Xiang Yu, Linda Liu, Mohamed Omar, and Raffay Hamid. Selective structured state-spaces for long-form video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6387–6397, 2023.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y Zhang, and JUN ZHOU. TimeMixer: Decomposable multiscale mixing for time series forecasting. In *International Conference on Learning Representations (ICLR)*, 2024a.

Xiao Wang, Shiao Wang, Yuhe Ding, Yuehang Li, Wentao Wu, Yao Rong, Weizhe Kong, Ju Huang, Shihao Li, Haoxiang Yang, et al. State space model for new-generation network alternative to transformers: A survey. *arXiv preprint arXiv:2404.09516*, 2024b.

Yunbo Wang, Mingsheng Long, Jianmin Wang, Zhifeng Gao, and Philip S Yu. PredRNN: Recurrent neural networks for predictive learning using spatiotemporal LSTMs. In *Advances in neural information processing systems*, 2017.

Zihan Wang, Fanheng Kong, Shi Feng, Ming Wang, Han Zhao, Daling Wang, and Yifei Zhang. Is Mamba effective for time series forecasting? *arXiv preprint arXiv:2403.11144*, 2024c.

Nicholas Watters, Daniel Zoran, Theophane Weber, Peter Battaglia, Razvan Pascanu, and Andrea Tacchetti. Visual Interaction Networks: Learning a physics simulator from video. *Advances in neural information processing systems*, 30, 2017.

Nicholas Watters, Loic Matthey, Christopher P Burgess, and Alexander Lerchner. Spatial Broadcast Decoder: A simple architecture for learning disentangled representations in vaes. *arXiv preprint arXiv:1901.07017*, 2019.

Paul J Werbos. Learning how the world works: Specifications for predictive networks in robots and brains. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics, NY*, 1987.

Paul J Werbos. Neural networks for control and system identification. In *Proceedings of the 28th IEEE Conference on Decision and Control,*, pp. 260–265. IEEE, 1989.

Christopher K I Williams and Michalis K Titsias. Greedy learning of multiple objects in images using robust statistics and factorial learning. *Neural Computation*, 16(5):1039–1062, 2004.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Advances in Neural Information Processing Systems*, 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 *International Conference on Learning Representations*, 2023a.

Ziyi Wu, Nikita Dvornik, Klaus Greff, Thomas Kipf, and Animesh Garg. SlotFormer: Unsupervised visual dynamics simulation with object-centric models. In *International Conference on Learning Representations*, 2023b.

Jing Nathan Yan, Jiatao Gu, and Alexander M Rush. Diffusion models without attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8239–8249, 2024.

Yuhuan Yang, Chaofan Ma, Jiangchao Yao, Zhun Zhong, Ya Zhang, and Yanfeng Wang. Remamber: Referring image segmentation with mamba twister. *arXiv preprint arXiv:2403.17839*, 2024a.

Zhichao Yang, Avijit Mitra, Sunjae Kwon, and Hong Yu. ClinicalMamba: A generative clinical language model on longitudinal clinical notes. *arXiv preprint arXiv:2403.05795*, 2024b.

Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B Tenenbaum. CLEVRER: Collision events for video representation and reasoning. In *International Conference on Learning Representations*, 2020.

Alan Yuille and Daniel Kersten. Vision as Bayesian inference: analysis by synthesis? *Trends in cognitive sciences*, 2006.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 11121–11128, 2023.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. OPT: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *International Conference on Learning Representations*, 2023.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International Conference on Machine Learning*, 2022.

A Related Works

From a historical perspective, "world models" or using models to learn environmental dynamics and leveraging them in policy training has an extensive literature, with early foundations laid in the 1980s using feed-forward neural networks (FNNs) (Werbos, 1987; Munro, 1987; Werbos, 1989; Robinson & Fallside, 1989; Nguyen & Widrow, 1990) and in the 1990s with RNNs (Schmidhuber, 1990b;a; 1991b;a). Notably, PILCO (Deisenroth & Rasmussen, 2011; McAllister & Rasmussen, 2016) has emerged as a key probabilistic model-based method, using Gaussian processes (GPs) (MacKay et al., 1998) to learn system dynamics from limited data and train controllers for tasks like pendulum swing-up and unicycle balancing. While GPs perform well with small, low-dimensional datasets, their computational complexity limits scalability in high-dimensional scenarios. To address this, later works (Gal et al., 2016; Depeweg et al., 2016) have adopted Bayesian neural networks (Kononenko, 1989), which have demonstrated success in control tasks with well-defined states (Hein et al., 2017). However, these methods remain limited when modeling high-dimensional environments, such as sequences of raw pixel frames. In the context of reinforcement learning, using recurrent models to learn system dynamics from compressed latent spaces has significantly improved data efficiency (Schmeckpeper et al., 2020; Finn et al., 2016). While the development of internal models for reasoning about future states using RNNs dates back to the early 1990s, subsequent works, such as "Learning to Think" (Schmidhuber, 2015) and "World Models" (Ha & Schmidhuber, 2018), have extended this by introducing RNN-based frameworks that model environments and reason about future outcomes. These RNN-based models have been applied to future frame generation (Chiappa et al., 2017; Oh et al., 2015; Denton et al., 2017) and reasoning about future outcomes (Silver et al., 2017; Watters et al., 2017). However, as RNNs suffer from the vanishing gradients problem (Hochreiter, 1991; Pascanu et al., 2013), recently there has been a growing interest in using Transformers (Chen et al., 2022; Robine et al., 2023; Micheli et al., 2022) and SSM-based appraoches (Gu & Dao, 2023; Hafner et al., 2023; Samsami et al., 2024) for world modeling.

As world modelling is fundamentally intertwined with sequence modelling (Schmidhuber, 1990b), it often carries temporal implications that align with the principle of history compression (Schmidhuber, 1992a; 2003). Temporal selectivity is essential in these models, with Recurrent Neural Networks (RNNs), particularly those with gating mechanisms like LSTMs (Hochreiter & Schmidhuber, 1997), GRUs (Cho, 2014), and xLSTMs (Beck et al., 2024), being well-suited for this task. However, learning from high-dimensional sequential data complicates the problem, posing a core challenge in spatial-temporal learning. This challenge is exacerbated by the quadratic computation scaling in transformers, despite their success. Approaches like dimensionality reduction (Hotelling, 1933; Tipping & Bishop, 1999; Kingma, 2013) and predictability minimisation (Schmidhuber, 1992b; Ghahramani, 1994) must adhere to the principle of history compression along the temporal axis, rather than compressing spatial information at each time step independently. From the perspective of information bottleneck principle (Tishby et al., 2000), the goal is to selectively extract the "bottleneck" from high-dimensional sequences that is most useful for world modelling tasks like predicting future events.

Recently, the emergence of Mamba (Gu & Dao, 2023) and other SSM-based frameworks (Gu et al., 2021; Dao & Gu, 2024; Wang et al., 2024b) has garnered widespread attention for their strong performance in efficient sequence modelling. Mamba structure is similar to LSTM (Hochreiter & Schmidhuber, 1997), in the sense that it utilises a forget gate, an input gate, and an output gate. The key difference is that these gates depend only on the previous input (not on the hidden state representing the history of inputs so far). While this hinders their representation power (e.g., cannot solve the parity problem (Hochreiter & Schmidhuber, 1996; Schmidhuber et al., 2007; Srivastava et al., 2015)), this formulation enables parallel computation of selective history compression via sub-linear sequential attention (Dao & Gu, 2024), constructing dependencies between distant data points within the sequence. This sparks their successful applications across various tasks including language modelling (Mehta et al., 2022; Grazzi et al., 2024; He et al., 2024), deep noise suppression (Du et al., 2024b), and clinical note understanding (Yang et al., 2024b). Additionally, many SSM-based vision models have been proposed for tasks such as classification (Du et al., 2024a; Shi et al., 2024; Baron et al., 2023; Huang et al., 2024; Smith et al., 2023; Nguyen et al., 2022), detection (Chen et al., 2024), segmentation (Yang et al., 2024a; Ma et al., 2024), generation (Yan et al., 2024; Fei et al., 2024), and video understanding (Islam & Bertasius, 2022; Wang et al., 2023). Despite their success, existing SSMs often lack efficient mechanisms for handling high-dimensional spatial data, relying primarily on linear and rigid structural biases (Gu et al., 2021; Dao & Gu, 2024), particularly when dealing with permutable spatial structures.

Multivariate time-series forecasting (MTSF) and object-centric representation learning (OCRL) both involve working with noisy, high-dimensional sequential data, making them critical benchmarks for evaluating world models with significant real-world impact. We chose to assess our models on these tasks because they exemplify the primary challenge of spatial-temporal learning. Existing MTSF methods either struggle to effectively model long-term temporal dependencies (Salinas et al., 2020; Zhang & Yan, 2023; Nie et al., 2023) or fail to effectively leverage the cross-variate regularities in high-dimensional inputs (Wu et al., 2021; Zhou et al., 2022; Wu et al., 2023a; Zeng et al., 2023; Wang et al., 2024a), resulting in inaccuracies in future-state forecasting. Two notable models in the field, iTransformer (Liu et al., 2024) and S-Mamba (Wang et al., 2024c), also have limitations. iTransformer suffers from the quadratic scaling of transformers, making it difficult to capture long-term dependencies, while S-Mamba struggles with handling the spatial structures of the data. Object-centric representations, or "slots", are designed to capture "objects", i.e. solving the binding problem (Greff et al., 2020). Our goal is to capture modularities, or "factors", that remain invariant across sequences, framing the discovery of spatial regularities in history compression as another instance of the binding problem. Although a philosophical discussion on whether these "factors" should align with common-sense "objects" is beyond the scope of this paper, OCRL is closely related and serves as a good demonstration of our approach. OCRL originated from the *vision-as-inverse-Bayes* framework (Yuille & Kersten, 2006), initially applied to images (Burgess et al., 2019; Greff et al., 2019; Locatello et al., 2020), later extended to videos (Nanbo et al., 2020; Kipf et al., 2023), and developed into object-centric world models (Lin et al., 2020; Kipf et al., 2019; Wu et al., 2023b; Stanić et al., 2023). Recent OCRL works heavily rely on the Slot Attention (SA) mechanism (Locatello et al., 2020) for object discovery, which is closely related to our routing modules. We view the SA, which also satisfies the LPE and RPI properties, as a suitable but computationally expensive alternative to equation 12.

B Proofs

Theorem (Restatement of Theorem 1). **FACTS** *as defined in equation 20 is L.P.E. and R.P.I.*

Proof. Let $\sigma_Z \in S_k$, $\sigma_X^1, \sigma_X^2, \ldots, \sigma_X^t \in S_m$, $Z_0, Z_t \in \mathbb{R}^{k\times d}$, $X_1, X_2, \ldots, X_t \in \mathbb{R}^{m\times d}$. By equation 17, equation 18 equation 19, and equation 20, it is sufficient to show

$$\sigma_Z\, \mathbf{FACTS}(Z_k, Z_0, X_k) = \mathbf{FACTS}(\sigma_Z Z_k, \sigma_Z Z_0, \sigma_X^t X_k).$$

for all $k \in \mathbb{N}, k \in [1, t]$.

$$\begin{aligned}
&\sigma_Z\, \mathbf{FACTS}(Z_k, Z_0, X_k) \\
=&\sigma_Z(\bar{A}(Z_0, X_k) \odot Z_{k-1} + \bar{B}(Z_0, X_k) \odot U(Z_0, X_k)) \\
=&\sigma_Z \bar{A}(Z_0, X_k) \odot \sigma_Z Z_{k-1} + \sigma_Z \bar{B}(Z_0, X_k) \odot \sigma_Z U(Z_0, X_k).
\end{aligned}$$

Since $\bar{\boldsymbol{A}}, \bar{\boldsymbol{B}}, \boldsymbol{U}$ are L.P.E. and R.P.I.,

$$\begin{aligned}
&\sigma_Z\, \mathbf{FACTS}(Z_k, Z_0, X_k) \\
=&\sigma_Z \bar{A}(Z_0, X_k) \odot \sigma_Z Z_{k-1} + \sigma_Z \bar{B}(Z_0, X_k) \odot \sigma_Z U(Z_0, X_k) \\
=&\bar{A}(\sigma_Z Z_0, \sigma_X^k X_k) \odot \sigma_Z Z_{k-1} + \bar{B}(\sigma_Z Z_0, \sigma_X^k X_k) \odot U(\sigma_Z Z_0, \sigma_X^k X_k) \\
=&\,\mathbf{FACTS}(\sigma_Z Z_k, \sigma_Z Z_0, \sigma_X^t X_k).
\end{aligned}$$

□

Theorem (Restatement of Theorem 2). *if $\bar{\boldsymbol{A}}, \bar{\boldsymbol{B}}, \boldsymbol{U}$ are L.P.E. and R.P.I. functions of Z_{t-1} and X_t, any dynamics governed by equation 10 is L.P.E. and R.P.I.*

Algorithm 1 FACTS Encoder: an Implementation

```
Input: $X_{1:t} \in \mathbb{R}^{t \times m \times d}$                                ▷ $t$-sequential axis, $m$-spatial axis
Output: $Z_{1:t} \in \mathbb{R}^{t \times k \times d}$
Init: $Z_0 \in \mathbb{R}^{k \times d} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
Param: $\boldsymbol{A} = \alpha \in \mathbb{R}$

$X_{1:t} \leftarrow$ Conv2D(Rearrange$(X_{1:t}, [d, t, m])$; kernel=(1, 1))                    ▷ Input projection
$(X_{1:t}, \Delta_{1:t}) \leftarrow$ Conv2D$(X_{1:t}$; kernel=(dconv, 1)).split(axis=1)          ▷ Conv. along the $t$-axis
$(X_{1:t}, \boldsymbol{B}_{1:t}, \boldsymbol{C}_{1:t}) \leftarrow$ Silu(Conv2D$(X_{1:t}$; kernel $= (1, 1)$)).split(axis=1)
$\boldsymbol{U}_{1:t} \leftarrow$ Rout$(Z_0, X_{1:t})$                                        ▷ Routing for factorisation
$(\Delta_{1:t}, \boldsymbol{B}_{1:t}, \boldsymbol{C}_{1:t}) \leftarrow$ Rout$(Z_0, (X_{1:t}, \Delta_{1:t}, \boldsymbol{B}_{1:t}, \boldsymbol{C}_{1:t}))$          ▷ Routing for the SSM Params
$\Delta_{1:t}$ = Softplus$(\Delta_{1:t})$
$(\bar{\boldsymbol{A}}_{1:t}, \bar{\boldsymbol{B}}_{1:t}) \leftarrow$ Discretisation$(\Delta_{1:t}, \boldsymbol{A}_{1:t}, \boldsymbol{B}_{1:t})$
$Z_{1:t} \leftarrow$ StateSpacePropagation$(Z_0, \boldsymbol{U}_{1:t}, \bar{\boldsymbol{A}}_{1:t}, \bar{\boldsymbol{B}}_{1:t})$          ▷ $Z_{1:t} \in \mathbb{R}^{t \times k \times d}$, c.f. equation 17-19
$\hat{Z}_{1:t} \leftarrow \boldsymbol{C}_{1:t} \odot Z_{1:t}$                          ▷ $\hat{Z}_{1:t} \in \mathbb{R}^{t \times k \times d}$ for selective output, e.g. Dec$(\hat{Z}_{1:t})$

Return: $\hat{Z}_{1:t}, Z_{1:t}$                                ▷ $\hat{Z}_{1:t}$: the output; $Z_{1:t}$: the state representation
```

Proof. Let $Z_{t-1} \in \mathbb{R}^{k \times d}, X_t \in \mathbb{R}^{m \times d}, \sigma_Z \in S_k$, and $\sigma_X \in S_m$ be matrices. Assume $\bar{\boldsymbol{A}}, \bar{\boldsymbol{B}}, \boldsymbol{U}$ are L.P.E. and R.P.I. functions of Z_{t-1} and X_t. By expanding equation 10,

$$\bar{A}_t(\sigma_Z Z_{t-1}, \sigma_X X_t) \odot \sigma_Z Z_{t-1} + \bar{B}_t(\sigma_Z Z_{t-1}, \sigma_X X_t) \odot U_t(\sigma_Z Z_{t-1}, \sigma_X X_t) \tag{21}$$

$$= \sigma_Z \bar{A}_t(Z_{t-1}, X_t) \odot \sigma_Z Z_{t-1} + \sigma_Z \bar{B}_t(Z_{t-1}, X_t) \odot \sigma_Z U_t(Z_{t-1}, X_t) \tag{22}$$

$$= \sigma_Z(\bar{A}_t(Z_{t-1}, X_t) \odot Z_{t-1} + \bar{B}_t(Z_{t-1}, X_t) \odot Z_{t-1} \odot U_t(Z_{t-1}, X_t)) \tag{23}$$

$$= \sigma_Z Z_t \tag{24}$$

□

C Implementation Details

C.1 FACTS Architecture

We provide an implementation of FACTS-based encoder in Algorithm 1, which is used in all our experiments.

C.2 Long-Term Forecasting

Datasets We use a collection of 9 widely-adopted public datasets, specifically designed for multivariate time series forecasting (MTSF) tasks, as presented in Table 5. These datasets span various domains, including traffic monitoring, electricity consumption, weather forecasting, and solar energy production, offering a diverse range of variates and time granularities. Notable datasets include the traffic data, electricity and solar energy for energy-related forecasting, and the ETT datasets for electric power and transformer temperature forecasting. The temporal resolutions of the datasets range from minutes to days, making them ideal for evaluating the performance of forecasting models over both short-term and long-term horizons.

Baselines

- *S-Mamba* (Wang et al., 2024c): This baseline adapts Mamba models for MTS data by utilising a bidirectional scan on variates, achieving superior results compared to the previous leading method, iTransformer.
- *iTransformer* (Liu et al., 2024): An inverted transformer architecture that captures univariate history and cross-variate dependencies within a look-back window, though limited by the quadratic scaling of transformers. iTransformer has been leading the long-term forecasting task
- *TimesNet* (Wu et al., 2023a): Specialises in modelling multi-periodicity and interactions among periodic signals in MTS data.

Table 5: The datasets for MTSF evaluation.

Datasets	Variates	Timesteps	Granularity
ETTm1 & ETTm2	7	17,420	15min
ETTh1 & ETTh2	7	69,680	1hour
Electricity	321	26,304	1hour
Traffic	862	17,544	1hour
Exchange	8	7,588	1day
Weather	21	52,696	10min
Solar-Energy	137	52,560	10min

- *CrossFormer* (Zhang & Yan, 2023): The emphasis is on modelling cross-dimension (spatial) interactions within MTS data.
- *PatchTST* (Nie et al., 2023): Uses patching techniques to segment sub-time sequences and model channel-wise transitions, improving temporal modelling.

FACTS for MTSF Due to the noisy nature of raw input data, a single time step (represented as a multivariate vector) often carries limited meaningful information. A common approach to handle this is to introduce feature encoders to pre-process the data, e.g. encoding, adding temporal and positional embeddings-as used in the baselines. We employ a set encoder to map the input multivariate sequences of size $t \times m$ into $t \times m \times d$, augmenting the tensor with an additional dimension that allows each time step to be represented as a set of m features, each of d-dimensional size. This resulting tensor, $t \times m \times d$, serves as the direct input to the FACTS model. For prediction, we adhere to the standard practice in the TimeLib benchmark, which treats time-series models as encoders designed for single-step predictions, rather than auto-regressive forecasting. We show the MTSF process of FACTS in Algorithm 2. Note that our decoder, namely "factor-graph decoder", is crucially designed to be invariant to the permutation of factors and processes the latent factors in parallel. Each factor independently makes predictions without relying on others. Specifically, the decoder aggregates the individual predictions of each factor by applying a softmax-weighted sum, ensuring that the final prediction effectively combines the contributions of all factors while maintaining permutation invariance. We details the aforementioned modules, i.e. the three embedders and the factor-graph decoder in the following list:

Algorithm 2 FACTS for Multivariate Time Series Forecasting

```
Input: $\boldsymbol{x}_{1:t} \in \mathbb{R}^{t \times m}$
Output: $\boldsymbol{x}_{t+1:t+f} \in \mathbb{R}^{f \times m}$

$X_{1:t} \leftarrow$ SetEncoder($\boldsymbol{x}_{1:t}$)    ▷ $X_{1:t} \in \mathbb{R}^{t \times m \times d}$, pre-processing for FACTS
$Z_{1:t} \leftarrow \mathbf{FACTS}(X_{1:t})$    ▷ $Z_{1:t} \in \mathbb{R}^{t \times k \times d}$, see Algorithm 1 for $\mathbf{FACTS}()$
$Z_{t+1:t+f} \leftarrow$ Predictor($Z_{1:t}$)    ▷ $Z_{t+1:t+f} \in \mathbb{R}^{f \times k \times d}$, where $f$ is the prediction length
$\boldsymbol{x}_{t+1:t+f} \leftarrow$ FactorGraphDecoder($Z_{t+1:t+f}$)    ▷ $\boldsymbol{x}_{t+1:t+f} \in \mathbb{R}^{f \times m}$
$\boldsymbol{x}_{t+1:t+f} \leftarrow$ PostProcessing($\boldsymbol{x}_{t+1:t+f}$)

Return: $\boldsymbol{x}_{t+1:t+f}$
```

- **Discrete-Fourier Transform (DFT) decomposer**: this embedder applies the Fast Fourier Transform to each variate and decompose it into multiple spectral components. The top-k low-frequency signals are treated as the "trend component", while the high frequency signals represent the "seasonal component". By concatenating the "trend" and "seasonal" components, we embed each univariate signal at each time point into a 2D (or $k+1$ dimensional, if the top-k frequencies are not combined) vector representation, resulting in a set output that is compatible with FACTS.
- **Conv2d embedder**: this embedder applies a convolution along the temporal axis of the input data to aggregate information from nearby time points (i.e., "local context", with the look-back window controlling the kernel size. We implement this using PyTorch's standard Conv2d

module, setting the kernel size to $[\text{lookback}, 1]$ and padding with zeros at the initial time steps. Although such embedders learn the input-feature mapping directly from data, making them flexible in capturing different relationships in different datasets, the choice of look-back window size is often intuitive.

- **Multi-scale Conv2d (MS-Conv2d) Embedder**: this embedder retains the learning flexibility of a Conv2d embedder while extending the single look-back window to multiple scales. By combining different scales, it captures features at varying granularities, making it the most robust among the three embedders and consistently delivering strong results (as shown in Table 3).
- **Factor Graph Decoder (FGD)**: this decoder takes in the a set of predicted latents $Z_f = \{\boldsymbol{z}_f^i\}_{i=1:k}$ (c.f. Algorithm 2 for its definition) and first project each $\boldsymbol{z}_f^i \in \mathbb{R}^d$ (can run parallel) to $\tilde{\alpha}_f^i \in \mathbb{R}^m$ (the logits) and $\tilde{\boldsymbol{x}}_f^i \in \mathbb{R}^m$ (the prediction of the i-th factor). Then the k logits, which correspond to the k factor predictions, will be processed to k categorical probabilities by a soft-max function: $\alpha_f^i = softmax(\tilde{\alpha}_f^i, \{\tilde{\alpha}_f^j\}_{j\neq i})$. The output prediction is the weighted sum of the factor predictions w.r.t. their corresponding probabilities as: $\boldsymbol{x}_f = \sum_i^k \alpha_f^i \odot \tilde{\boldsymbol{x}}_f^i$, similar to the processes of spatial mixing and alpha blending in the vision and graphics communities (Porter & Duff, 1984; Williams & Titsias, 2004; Greff et al., 2017). In vision tasks, the Spatial Broadcast Decoder (SBD, Watters et al. (2019)), which shares similar properties with FGD but is more computationally expensive, is a more commonly used option.

C.3 Object-Centric World Modelling

Datasets We conducted both the *slot dynamics prediction* and *unsupervised object discovery* experiments on the CLEVRER dataset (Yi et al., 2020; Johnson et al., 2017), which contains 15k scenes featuring multiple moving objects such as spheres, cubes, and cylinders. We follow the setup in (Wu et al., 2023b), filtering out video clips with new objects entering the scene during the rollout period to ensure a consistent evaluation setting. For the slot dynamics prediction task, the input to FACTS is the latent object representations extracted using a pre-trained object-slot encoder (SAVi, Kipf et al. (2023)) from video frames. That is, we consider the output of the pre-trained SAVi our data in such task – same as (Wu et al., 2023b).

Setup As mentioned, for the slot dynamics prediction task, we extract the input latent object representations using a pre-trained object-slot encoder (SAVi Kipf et al. (2023)) from video frames, and train an auto-regressive rollouter with a single FACTS layer to predict the latent representations for the next 10 frames, based on the latent codes from 6 observed frames. During testing, to ensure a fair comparison with SlotFormer, we burn-in the first 6 frames and roll out (predict) 48 frames. The predicted object representations are visualised using a pre-trained, frozen SAVi decoder (a spatial broadcast decoder, Wu et al. (2023b)) to render video frames.

D Additional Results

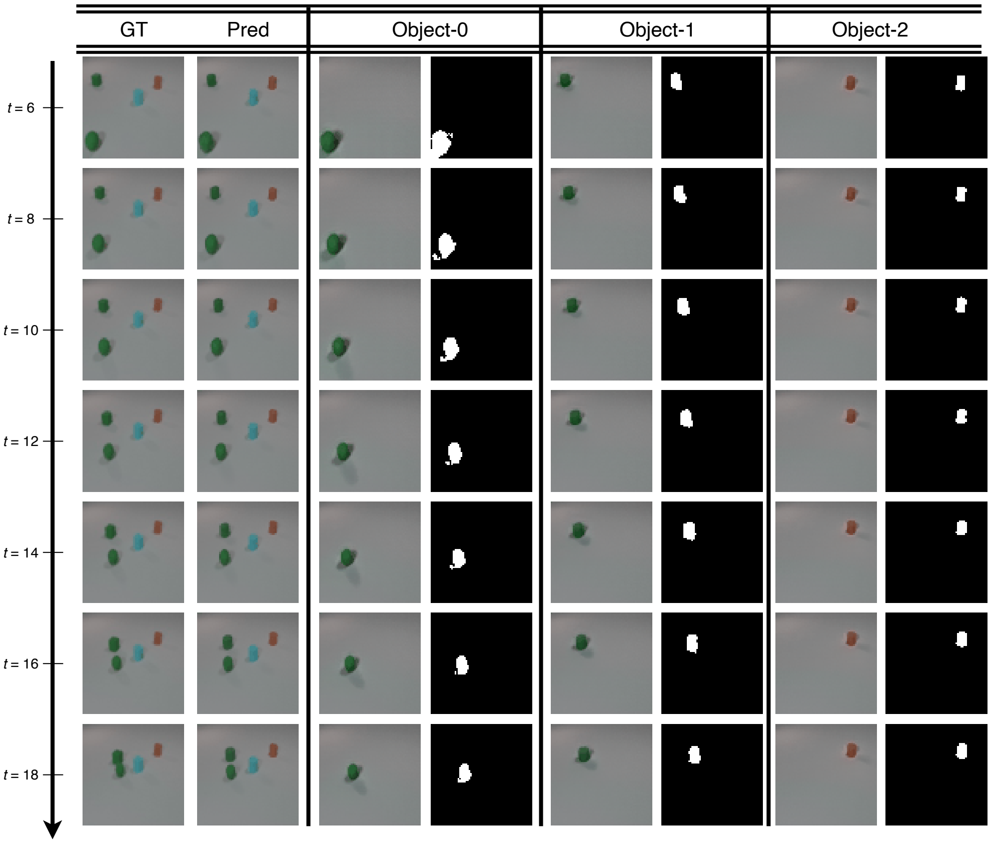


Figure 3: Qualitative results of slot dynamics prediction. The first column shows selected ground truth future frames, while the second column presents our predicted future slot-state rendered as frames. We show 3 object-centric dynamics in the remaining columns: two columns for each object: the left displays the predicted object dynamics, and the right shows the corresponding object masks.

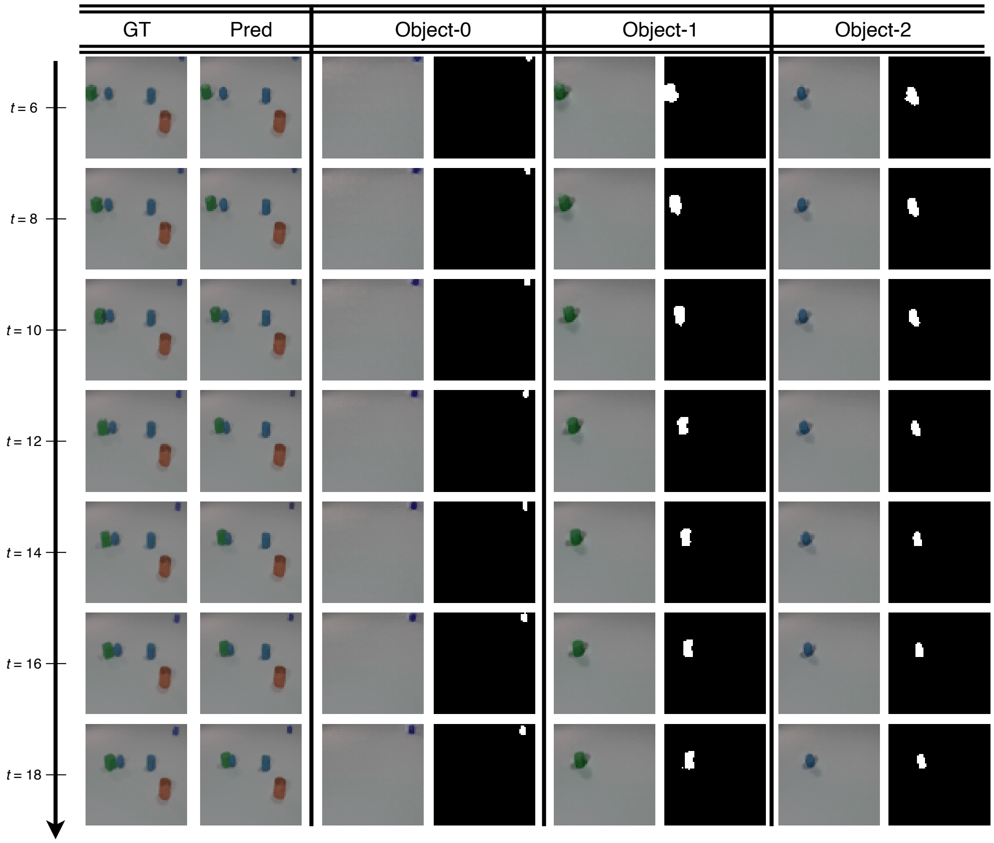


Figure 4: Qualitative results of slot dynamics prediction. The first column shows selected ground truth future frames, while the second column presents our predicted future slot-state rendered as frames. We show 3 object-centric dynamics in the remaining columns: two columns for each object: the left displays the predicted object dynamics, and the right shows the corresponding object masks.

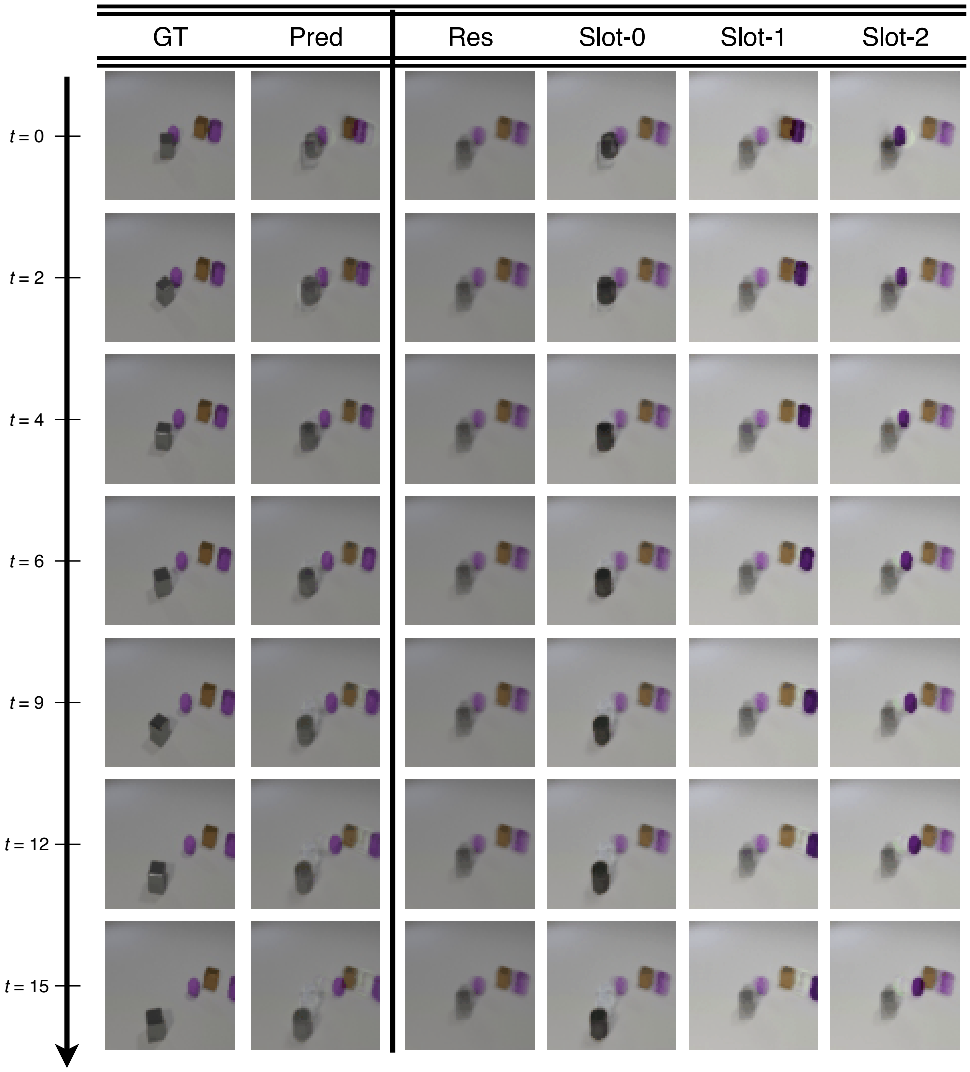


Figure 5: Qualitative results of unsupervised object discovery. The first column shows selected ground truth frames, with the first four (time-stamped < 6) as burn-in frames. The second column displays our future video frame predictions. The last three columns, i.e. the "slots", captures the discovered moving objects (with their dynamic movements shown in the respective columns) and are highlighted with dark colours for clarity. The "Residual" (Res) reconstructs the last burn-in frame for visual reference.

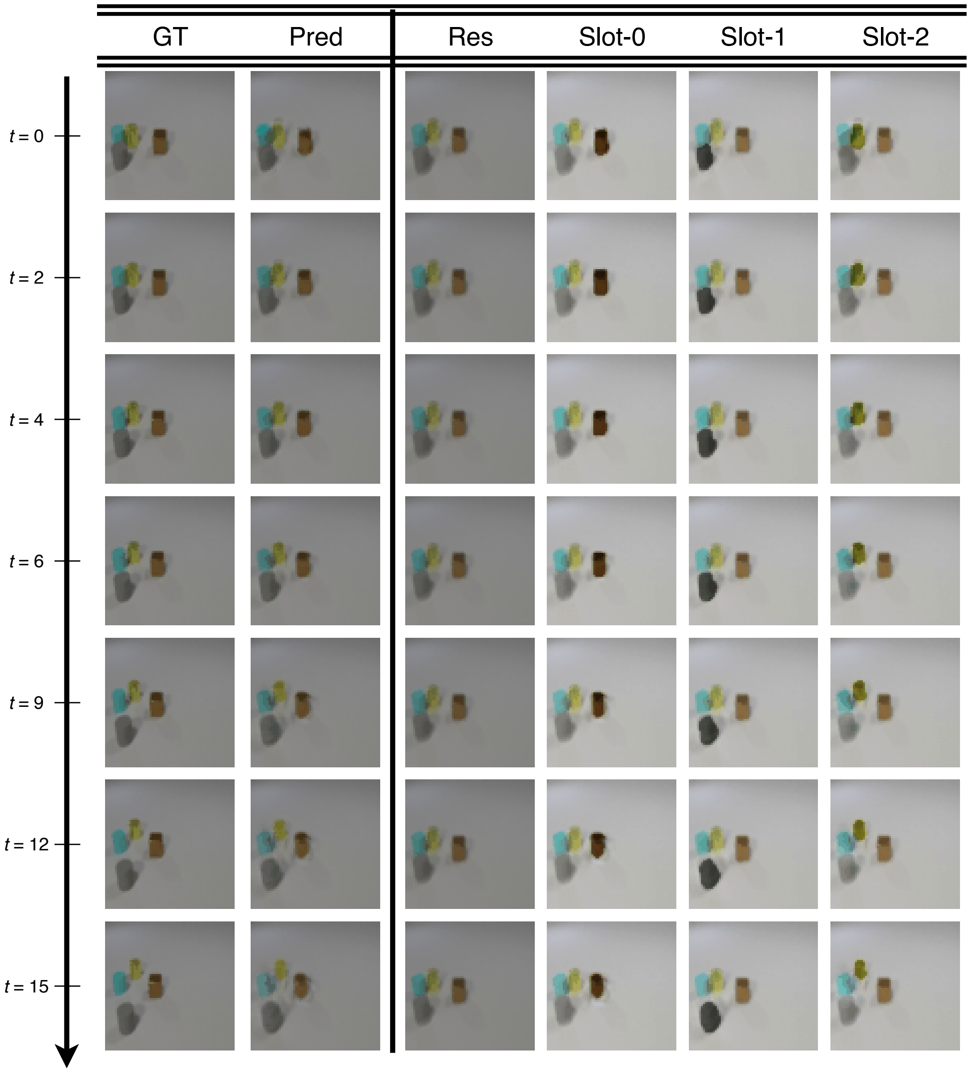


Figure 6: Qualitative results of unsupervised object discovery. The first column shows selected ground truth frames, with the first four (time-stamped < 6) as burn-in frames. The second column displays our future video frame predictions. The last three columns, i.e. the “slots”, captures the discovered moving objects (with their dynamic movements shown in the respective columns) and are highlighted with dark colours for clarity. The “Residual” (Res) reconstructs the last burn-in frame for visual reference.

Data		**FACTS (Ours)**		S-Mamba		iTransformer		TimesNet		PatchTST		DLinear		Crossformer		FEDformer		Autoformer	
	Pred.Len.	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	**0.326**	**0.363**	0.333	0.368	0.334	0.368	0.338	0.375	0.329	0.367	0.345	0.372	0.404	0.426	0.379	0.419	0.505	0.475
	192	**0.366**	0.386	0.376	0.390	0.377	0.391	0.374	0.387	0.367	**0.385**	0.380	0.389	0.450	0.451	0.426	0.441	0.553	0.496
	336	0.412	**0.407**	0.408	0.413	0.426	0.420	0.410	0.411	**0.399**	0.410	0.413	0.413	0.532	0.515	0.445	0.459	0.621	0.537
	720	0.468	0.441	0.475	0.448	0.491	0.459	0.478	0.450	**0.454**	**0.439**	0.474	0.453	0.666	0.589	0.543	0.490	0.671	0.561
	Avg.	0.393	**0.399**	0.398	0.405	0.407	0.410	0.400	0.406	**0.387**	0.400	0.403	0.407	0.513	0.496	0.448	0.452	0.588	0.617
ETTm2	96	**0.175**	**0.258**	0.179	0.263	0.180	0.264	0.187	0.267	**0.175**	0.259	0.193	0.292	0.287	0.366	0.203	0.287	0.255	0.339
	192	**0.241**	**0.300**	0.250	0.309	0.250	0.309	0.249	0.309	**0.241**	0.302	0.284	0.362	0.414	0.492	0.269	0.328	0.281	0.340
	336	**0.304**	**0.341**	0.312	0.349	0.311	0.348	0.321	0.351	0.305	0.343	0.369	0.427	0.597	0.542	0.325	0.366	0.339	0.372
	720	0.406	**0.400**	0.411	0.406	0.412	0.407	0.408	0.403	**0.402**	**0.400**	0.554	0.522	1.730	1.042	0.421	0.415	0.433	0.432
	Avg.	**0.281**	**0.325**	0.288	0.332	0.288	0.332	0.291	0.333	**0.281**	0.326	0.350	0.401	0.757	0.610	0.305	0.349	0.327	0.371
ETTh1	96	0.382	**0.390**	0.386	0.405	0.386	0.405	0.384	0.402	0.414	0.419	0.386	0.400	0.423	0.448	**0.376**	0.419	0.449	0.459
	192	0.433	**0.419**	0.443	0.437	0.441	0.436	0.436	0.429	0.460	0.445	0.437	0.432	0.471	0.474	**0.420**	0.448	0.500	0.482
	336	0.474	**0.440**	0.489	0.468	0.487	0.458	0.491	0.469	0.501	0.466	0.481	0.459	0.570	0.546	**0.459**	0.465	0.521	0.496
	720	**0.473**	**0.462**	0.502	0.489	0.503	0.491	0.521	0.500	0.500	0.488	0.519	0.516	0.653	0.621	0.506	0.507	0.514	0.512
	Avg.	0.441	**0.428**	0.455	0.450	0.454	0.447	0.458	0.450	0.469	0.454	0.456	0.452	0.529	0.522	**0.440**	0.460	0.496	0.487
ETTh2	96	**0.288**	**0.337**	0.296	0.348	0.297	0.349	0.340	0.374	0.302	0.348	0.333	0.387	0.745	0.584	0.358	0.397	0.346	0.388
	192	**0.374**	**0.392**	0.376	0.396	0.380	0.400	0.402	0.414	0.388	0.400	0.477	0.476	0.877	0.656	0.429	0.439	0.456	0.452
	336	**0.420**	**0.429**	0.424	0.431	0.428	0.432	0.452	0.452	0.426	0.433	0.594	0.541	1.043	0.731	0.496	0.487	0.482	0.486
	720	**0.422**	**0.439**	0.426	0.444	0.427	0.445	0.462	0.468	0.431	0.446	0.831	0.657	1.104	0.763	0.463	0.474	0.515	0.511
	Avg.	**0.376**	**0.399**	0.381	0.405	0.383	0.407	0.414	0.427	0.387	0.407	0.559	0.515	0.942	0.684	0.437	0.449	0.450	0.459
Electricity	96	0.143	0.240	**0.139**	**0.235**	0.148	0.240	0.168	0.272	0.181	0.270	0.197	0.282	0.219	0.314	0.193	0.308	0.201	0.317
	192	**0.158**	0.255	0.159	0.255	0.162	**0.253**	0.184	0.289	0.188	0.274	0.196	0.285	0.231	0.322	0.201	0.315	0.222	0.334
	336	**0.171**	**0.268**	0.176	0.272	0.178	0.269	0.198	0.300	0.204	0.293	0.209	0.301	0.246	0.337	0.214	0.329	0.231	0.338
	720	**0.198**	**0.293**	0.204	0.298	0.225	0.317	0.220	0.320	0.246	0.324	0.245	0.333	0.280	0.363	0.246	0.355	0.254	0.361
	Avg.	**0.168**	**0.264**	0.170	0.265	0.178	0.270	0.192	0.295	0.205	0.290	0.212	0.300	0.244	0.334	0.214	0.327	0.227	0.338
Exchange	96	**0.081**	**0.197**	0.086	0.207	0.086	0.206	0.107	0.234	0.088	0.205	0.088	0.218	0.256	0.367	0.148	0.278	0.197	0.323
	192	**0.172**	**0.295**	0.182	0.304	0.177	0.299	0.226	0.344	0.176	0.299	0.176	0.315	0.470	0.509	0.271	0.315	0.300	0.369
	336	0.322	0.407	0.332	0.418	0.331	0.417	0.367	0.448	**0.301**	**0.397**	0.313	0.427	1.268	0.883	0.460	0.427	0.509	0.524
	720	0.846	0.692	0.867	0.703	0.847	**0.691**	0.964	0.746	0.901	0.714	**0.839**	0.695	1.767	1.068	1.195	0.695	1.447	0.941
	Avg.	0.355	**0.398**	0.367	0.408	0.360	0.403	0.416	0.443	0.367	0.404	**0.354**	0.414	0.940	0.707	0.519	0.429	0.613	0.539
Traffic	96	0.444	0.285	**0.382**	**0.261**	0.395	0.268	0.593	0.321	0.462	0.295	0.650	0.396	0.522	0.290	0.587	0.366	0.613	0.388
	192	0.455	0.289	**0.396**	**0.267**	0.417	0.276	0.617	0.336	0.466	0.296	0.598	0.370	0.530	0.293	0.604	0.373	0.616	0.382
	336	0.471	0.299	**0.417**	**0.276**	0.433	0.283	0.629	0.336	0.482	0.304	0.605	0.373	0.558	0.305	0.621	0.383	0.622	0.337
	720	0.511	0.320	**0.460**	**0.300**	0.467	0.302	0.640	0.350	0.514	0.322	0.645	0.394	0.589	0.328	0.626	0.382	0.660	0.408
	Avg.	0.470	0.298	**0.414**	**0.276**	0.428	0.282	0.620	0.336	0.481	0.304	0.625	0.383	0.550	0.304	0.610	0.376	0.628	0.379
Weather	96	0.164	0.211	0.165	**0.210**	0.174	0.214	0.172	0.220	0.177	0.218	0.196	0.255	**0.158**	0.230	0.217	0.296	0.266	0.336
	192	0.213	0.254	0.214	**0.252**	0.221	0.254	0.219	0.261	0.225	0.259	0.237	0.296	**0.206**	0.277	0.276	0.336	0.307	0.367
	336	**0.271**	**0.295**	0.274	0.297	0.278	0.296	0.280	0.306	0.278	0.297	0.283	0.335	0.272	0.335	0.339	0.380	0.359	0.395
	720	0.352	0.347	0.350	**0.345**	0.358	0.347	0.365	0.359	0.354	0.348	**0.345**	0.381	0.398	0.418	0.403	0.428	0.419	0.428
	Avg.	**0.250**	0.277	0.251	**0.276**	0.258	0.278	0.259	0.287	0.259	0.281	0.265	0.317	0.259	0.315	0.309	0.360	0.338	0.382
Solar-Energy	96	**0.203**	0.238	0.205	0.244	**0.203**	**0.237**	0.250	0.292	0.234	0.286	0.290	0.378	0.310	0.331	0.242	0.342	0.884	0.711
	192	0.253	0.271	0.237	0.270	**0.233**	**0.261**	0.296	0.318	0.267	0.310	0.320	0.398	0.734	0.725	0.285	0.380	0.834	0.692
	336	0.276	0.291	0.258	0.288	**0.248**	**0.273**	0.319	0.330	0.290	0.315	0.353	0.415	0.750	0.735	0.282	0.376	0.941	0.723
	720	0.292	0.297	0.260	0.288	**0.249**	**0.275**	0.338	0.337	0.289	0.317	0.356	0.413	0.769	0.765	0.357	0.427	0.882	0.717
	Avg.	0.256	0.274	0.240	0.273	**0.233**	**0.262**	0.301	0.319	0.270	0.307	0.330	0.401	0.641	0.639	0.291	0.381	0.885	0.711

Table 6: Full results for the MTS long-term forecasting task (in MSE↓ and MAE↓). We compare extensive competitive models under different prediction lengths. The input sequence length is set to 36 for the ILI dataset and 96 for the others. Avg is averaged from all four prediction lengths. For each metric and each dataset, the top performance and the second best are highlighted in **red** and blue, respectively.