REGIMEUNFOLD: CONTINUAL REGIME-SPACE LEARNING FOR DEEP HIDDEN MARKOV PROCESS FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep hidden Markov process forecasting aims to predict future observations while explicitly identifying the latent regimes governing their dynamics. Existing approaches typically combine deep learning with Hidden Markov Models (HMMs) or their variants, seeking to retain the strong nonlinear predictive capability of deep models while preserving explicit latent regime representations. However, these methods generally operate over a fixed regime set, making their performance sensitive to a prespecified regime-space size and limiting adaptation after deployment. To address this limitation, we propose RegimeUnfold, a continual regime-space learning framework with offline regime discovery and online regime evolution for deep hidden Markov process forecasting. During the offline stage, RegimeUnfold jointly infers discrete regime assignments and learns predictive dynamics while progressively constructing and refining the regime space without predefining its size. During the online stage, it performs hard regime decoding and uses predictive-insufficiency and structural-novelty evidence to retain applicable regimes, update drifting regimes, or expand the regime space under stability constraints. Experiments on five synthetic and four real-world datasets demonstrate accurate forecasting, reliable regime inference, and effective adaptation to evolving predictive dynamics.

1 INTRODUCTION

Deep hidden Markov process forecasting requires not only accurate prediction of future observations but also explicit inference of the latent regimes governing their dynamics. When the number of regimes is unknown, fixing the regime space in advance may merge distinct predictive mechanisms or unnecessarily separate similar ones. Moreover, existing regimes may drift after deployment while previously unseen dynamics may emerge, requiring the regime space to be learned from data and continually adapted over time.

Latent regime discovery and regime-space adaptation have long been studied within the HMM framework. Classical HMM extensions model richer duration, dependency, and interaction structures (Lin et al., 2022; Rodriguez-Fernandez et al., 2017; Li & Zhang, 2023), while Bayesian nonparametric HMMs infer the effective number of latent states from data (Beal et al., 2001; Fox et al., 2008a; Dawson et al., 2017). More recent methods further support online state-space expansion as streaming observations arrive (Chakraborty et al., 2023; Yiu et al., 2026). These models retain explicit discrete latent regimes and provide interpretable representations of regime dynamics. However, their observation models typically rely on parametric regime-conditional distributions, switching linear or autoregressive models, or Gaussian processes, which may be insufficient for capturing complex nonlinear predictive dependencies in high-dimensional multivariate time series. Their regime transitions are also generally governed by predefined Markovian dependence structures, limiting the modeling of richly history-dependent, input-dependent, or time-varying transition patterns.

To capture complex nonlinear observation patterns and predictive dependencies while retaining explicit latent-regime dynamics, researchers have increasingly combined deep learning with HMMs. Existing methods generally enhance neural emission modeling while preserving HMM-style tran-

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

sitions (Dahl et al., 2012; Ilhan et al., 2023; Bansal & Zhou, 2025), or jointly parameterize transition and emission potentials with neural networks while retaining likelihood-based HMM learning and probabilistic inference (Tran et al., 2016; Song et al., 2023). A forecasting-oriented study has instead explored a distribution-free framework for regime-aware deep hidden Markov process forecasting (Jiang et al., 2026). Despite improved expressiveness, several limitations remain for forecasting. First, most DL-HMM hybrids are trained with likelihood-based objectives that are not directly aligned with forecasting accuracy, since higher marginal likelihood does not necessarily imply better out-of-sample prediction (Lotfi et al., 2022). Second, most DL-HMM hybrids rely on posterior-weighted soft decoding, which may blur regime boundaries, weaken regime-specific specialization, and produce forecasts that do not correspond to any individual predictive mechanism. Most importantly, existing deep HMM forecasting methods generally assume a regime space that is specified before training and remains fixed thereafter. This makes performance sensitive to the prespecified regime-space size and prevents previously unseen predictive dynamics from being incorporated after deployment, forcing them into existing regimes and potentially degrading regime-conditioned forecasting. Notably, directly extending nonparametric or online HMM mechanisms to deep forecasting is also nontrivial. Beyond expanding the discrete regime space, a newly emerging regime must be identified and validated from limited streaming evidence, modeled with a corresponding nonlinear predictive mechanism, and incorporated without disrupting previously learned regime-specific predictors.

To address these limitations, we propose RegimeUnfold, a continual regime-space learning framework for deep hidden Markov process forecasting that explicitly models discrete latent regimes and regime-conditioned predictive dynamics. RegimeUnfold operates through offline regime discovery and online regime evolution. The offline stage constructs and refines the latent regime space without predefining the number of regimes, while the online stage performs hard regime decoding and continually updates or expands the learned regime space when needed. In this way, RegimeUnfold enables nonlinear forecasting and explicit regime inference while continually adapting the latent regime space as predictive dynamics evolve. The main contributions are summarized as follows:

- We propose RegimeUnfold, a continual regime-space learning framework for deep hidden Markov process forecasting that integrates nonlinear forecasting with explicit discrete regime inference without assuming a fixed regime space.
- We develop an offline regime discovery mechanism that constructs and refines the latent regime space without predefining the number of regimes. It jointly infers discrete regime assignments and learns predictive dynamics through iterative regime differentiation, structural reorganization, and predictive refinement.
- We develop an online regime evolution mechanism that continually adapts the learned regime space as predictive dynamics evolve. It performs hard regime decoding over the current regime space and uses predictive-insufficiency and structural-novelty evidence to assess whether existing regimes remain applicable and, when necessary, update drifting regimes or expand the regime space under stability constraints.

2 Related Work

Classical and Extended HMMs. HMMs are foundational probabilistic models for sequential data, providing explicit latent-regime representations for temporal dynamics (Mor et al., 2021). Classical extensions relax different structural assumptions of the basic HMM by modeling higher-order state dependencies (Rodriguez-Fernandez et al., 2017), interactions among multiple state sequences (Wang et al., 2019), and non-exponential state durations (Lin et al., 2022). However, these models typically require the number of latent states to be specified in advance, which is difficult when the underlying regime structure is unknown. Bayesian nonparametric HMMs address this limitation by inferring the effective number of latent states from data (Beal et al., 2001; Fox et al., 2008a). Subsequent studies improve state persistence (Fox et al., 2008a;b), inference and state-space refinement (Van Gael et al., 2008; Tripuraneni et al., 2015; Hughes et al., 2015), and extend flexible-state models through discriminative objectives (Zhang et al., 2014), multi-scale dynamics (Saeedi et al., 2016), and structured relationships among latent states (Dawson et al., 2017). More recently, nonparametric HMMs have been extended toward sequential and online state-space adaptation. Chakraborty et al. (2023) combine sequential nonparametric state inference with Gaussian-

process dynamics to expand the latent state space while learning nonlinear state-specific dynamics from streaming data, while Yiu et al. (2026) improve the robustness of online regime discovery against transient outliers. It is worth noting that these approaches still rely on prescribed probabilistic structures for both observation and regime-transition modeling. For observation modeling, regime-specific dynamics are typically represented by parametric regime-conditional distributions, switching linear or autoregressive models, or Gaussian processes, which may be insufficient for complex nonlinear dependencies in high-dimensional multivariate time series. For regime-transition modeling, regime evolution generally follows predefined Markovian structures, making history-dependent, input-dependent, or time-varying transitions difficult to capture.

Deep Learning-HMM Hybrids. Deep sequence models are effective at capturing complex nonlinear temporal dependencies, but changing dynamics are often represented through continuous and implicit features rather than explicit discrete regimes (Lai et al., 2018; Gu & Dao, 2024). To combine nonlinear representation learning with explicit latent-regime modeling, a growing body of research has integrated deep learning with HMM-based structures. CD-DNN-HMM (Dahl et al., 2012) uses a deep neural network to estimate posterior probabilities over HMM states while retaining the conventional HMM state-transition structure. Markovian RNNs (Ilhan et al., 2023) embed HMM-based regime switching into recurrent neural architectures to capture regime-dependent dynamics in nonstationary time series. DEN-HMM (Bansal & Zhou, 2025) replaces conventional regime-dependent emission distributions with deep neural networks to model complex regime- and time-dependent multivariate observations. NHMM (Tran et al., 2016) parameterizes emission and transition probabilities with neural networks while retaining forward-backward inference and marginal-likelihood-based learning. NCTRL (Song et al., 2023) combines a discrete Markov process for unknown nonstationarity with neural nonlinear representations and time-delayed latent causal dynamics. DRL-STAF (Jiang et al., 2026) further develops a distribution-free framework that combines deep nonlinear forecasting with explicit hidden-regime estimation through deep reinforcement learning. These methods improve the expressiveness of latent-regime models, but most remain likelihood-driven rather than directly optimized for forecasting accuracy and rely on posterior-weighted soft decoding, which may blur regime boundaries and weaken regime-specific specialization. More fundamentally, existing DL-HMM hybrids typically retain a fixed regime set rather than continually learning the regime space, leaving latent regime discovery and online expansion largely unaddressed. Extending nonparametric or online HMM mechanisms to deep forecasting is nontrivial because a newly introduced regime must also be modeled with corresponding regime-specific predictive dynamics and incorporated without disrupting previously learned regimes.

3 Problem Formulation

Consider a sequential process $\{\mathbf{x}_t\}_{t=1}^{T}$, where $\mathbf{x}_t \in \mathbb{R}^d$ denotes the observation at time t. Let $\mathbf{H}_t = (\mathbf{x}_{t-h+1}, \ldots, \mathbf{x}_t)$ denote the most recent history of length h. We assume that the observations are governed by a temporally dependent discrete latent regime process $\{z_t\}$, and that the next observation follows regime-conditioned predictive dynamics $\mathbf{x}_{t+1} = g_{z_{t+1}}(\mathbf{H}_t) + \boldsymbol{\epsilon}_{t+1}$, where g_k denotes the unknown data-generating dynamics associated with regime k and $\boldsymbol{\epsilon}_{t+1}$ denotes observation noise. Under this formulation, forecasting requires inferring an explicit regime assignment $\hat{z}_{t+1}$ from the available history and producing the corresponding regime-conditioned prediction $\hat{\mathbf{x}}_{t+1} = f_{\hat{z}_{t+1}}(\mathbf{H}_t)$, where f_k denotes the learned predictor associated with regime k.

Let $\mathcal{R}_t$ denote the latent regime space available at time t, with $K_t = |\mathcal{R}_t|$. We consider $\mathcal{R}_t$ as a continually learned latent regime space whose size is not specified in advance and may expand as previously unseen predictive dynamics emerge. The objective is therefore to achieve accurate forecasting and explicit regime inference while continually learning and adapting the latent regime space over time.

4 RegimeUnfold

RegimeUnfold realizes continual regime-space learning through offline regime discovery and online regime evolution. The offline stage jointly infers discrete regime assignments and learns regime-conditioned predictive dynamics while progressively constructing and refining the latent regime space. The resulting regime structure and predictive models provide the basis for subse-

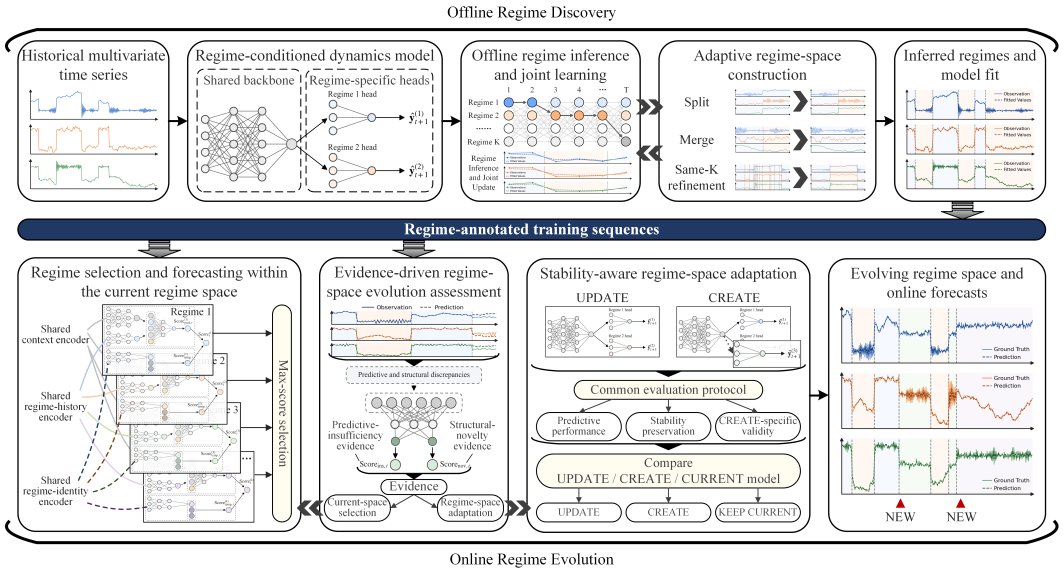


Figure 1: Overall structure of RegimeUnfold.

quent online inference and forecasting. The online stage performs hard regime decoding for regime-conditioned forecasting and accumulates predictive-insufficiency and structural-novelty evidence to assess whether the current regime space remains adequate. When adaptation is needed, it evaluates updating an existing regime or creating a new regime under stability constraints. Figure 1 illustrates the overall framework.

4.1 Offline Regime Discovery

The offline stage discovers discrete latent regimes and their predictive dynamics from unlabeled historical data without predefining the number of regimes. One-step prediction provides the learning signal, as observations from the same regime should exhibit similar predictive dynamics while different regimes should remain distinguishable. Because regime assignments, predictive dynamics, and regime-space structure are interdependent, RegimeUnfold alternates regime inference and predictive-model learning while progressively constructing and refining the latent regime space.

Regime-conditioned dynamics model. Given the observation history $\mathbf{H}_t$, a deep encoder produces a shared representation $\mathbf{h}_t = E_\theta(\mathbf{H}_t)$, which is further mapped to a low-dimensional regime code $\mathbf{c}_t = Q_\theta(\mathbf{h}_t)$, where θ collectively denotes the learnable parameters of the representation mappings. Based on the shared representation, RegimeUnfold decomposes prediction into common and regime-specific dynamics. For K active regimes, a shared base predictor f_0 captures the common predictive component, while each regime k has a lightweight residual predictor r_k and a regime-specific bias $\mathbf{b}_k$ to capture deviations from the shared dynamics. Each regime is also associated with a prototype $\mathbf{p}_k$ in the code space, which serves as a representation anchor for regime inference and structural comparison. The compatibility between the current regime code and regime k is measured by $d_{t,k} = \|\mathbf{c}_t - \mathbf{p}_k\|_2^2$.

To reduce interference when the observation history spans a regime transition, RegimeUnfold employs a guarded regime-specific temporal mask. The regime codes within the history are compared with the current regime prototypes to assess whether a reliable regime boundary is present. Earlier observations are downweighted for regime-specific residual prediction when such a boundary is detected, while the complete history is retained otherwise. Let $\widetilde{\mathbf{h}}_t^{(k)}$ denote the resulting regime-specific representation. The final one-step prediction under regime k is $\widehat{\mathbf{x}}_{t+1}^{(k)} = f_0(\mathbf{h}_t) + r_k(\widetilde{\mathbf{h}}_t^{(k)}) + \mathbf{b}_k$, where the shared base captures dynamics common across regimes, while the residual predictor and bias characterize regime-specific deviations.

Given hard regime assignments, the dynamics model is optimized using

$$\begin{aligned}\mathcal{L}_{\text{dyn}} =& \lambda_{\text{pred}}\mathcal{L}_{\text{pred}} + \lambda_{\text{base}}\mathcal{L}_{\text{base}} + \lambda_{\text{com}}\mathcal{L}_{\text{com}} + \lambda_{\text{sep}}\mathcal{L}_{\text{sep}} + \lambda_{\text{spec}}\mathcal{L}_{\text{spec}} \\ &+ \lambda_{\text{res}}\mathcal{L}_{\text{res}} + \lambda_{\text{bias}}\mathcal{L}_{\text{bias}} + \lambda_{\text{off}}\mathcal{L}_{\text{off}} + \lambda_{\text{mask}}\mathcal{L}_{\text{mask}}.\end{aligned} \tag{1}$$

Here, $\mathcal{L}_{\text{pred}}$ trains the assigned regime-conditioned predictor, while $\mathcal{L}_{\text{base}}$ keeps the shared predictor informative so that regime-specific residuals mainly capture deviations from common dynamics. $\mathcal{L}_{\text{com}}$ encourages regime codes to stay close to their assigned prototypes, and $\mathcal{L}_{\text{sep}}$ separates different regime prototypes in the code space. $\mathcal{L}_{\text{spec}}$ promotes regime specialization by encouraging the assigned predictor to outperform competing regime predictors. $\mathcal{L}_{\text{res}}$ and $\mathcal{L}_{\text{bias}}$ regularize the magnitudes of regime-specific residuals and biases, while $\mathcal{L}_{\text{off}}$ suppresses residual responses from non-assigned regimes. Finally, $\mathcal{L}_{\text{mask}}$ discourages unnecessary activation of the guarded temporal mask.

Offline regime inference and joint learning. For a candidate regime-space size K, assigning each sample independently to the regime with the smallest prediction error may make the inferred path sensitive to local prediction noise and lead to fragmented regime assignments. RegimeUnfold therefore combines predictive evidence, directional prediction bias, code-space compatibility, and temporal continuity to infer a complete hard regime path.

For each sample-regime pair, prediction error provides the primary assignment evidence. The squared prediction errors are first computed for individual output dimensions and then summarized using both their mean and maximum values. The mean measures the overall predictive fit across outputs, while the maximum prevents a large discrepancy in one output from being concealed by good predictions on the others. Because squared errors discard the direction of prediction deviations, RegimeUnfold further uses a signed prediction-bias term to distinguish systematic overprediction from underprediction. In addition, the prototype distance $d_{t,k}$ provides complementary representation-level evidence when different regimes produce similar prediction errors. These predictive, directional, and representation-level terms are normalized and combined into the local assignment cost $\psi_{t,k}$. It should be noted that local evidence alone may still yield fragmented regime paths. RegimeUnfold therefore decodes the complete hard regime sequence by combining the local assignment costs with switch and duration regularization:

$$\widehat{\mathbf{z}} = \arg\min_{\mathbf{z}} \left[\sum_{t} \psi_{t,z_{t+1}} + \kappa_{\text{sw}} \sum_{t} \mathbb{I}(z_{t+1} \neq z_t) + \sum_{\mathcal{G} \in \text{Seg}(\mathbf{z})} \phi(|\mathcal{G}|) \right], \tag{2}$$

where $\mathbb{I}(\cdot)$ is the indicator function, $\text{Seg}(\mathbf{z})$ denotes the set of contiguous regime segments, $|\mathcal{G}|$ denotes the length of segment $\mathcal{G}$, κ_{sw} penalizes unnecessary regime switches, and $\phi(\cdot)$ softly penalizes short segments. The resulting path is obtained by segment-level dynamic programming.

Regime assignments and predictive dynamics are optimized through a hard-EM-style alternating procedure. Given the current hard regime path, the dynamics model is first updated by optimizing the shared and regime-specific predictors under these assignments. The updated model is then fixed and used to recompute regime-wise predictive costs, from which a new hard regime sequence is decoded. The newly decoded path is subsequently used to update the dynamics model again, and this assignment-parameter update cycle is repeated until the solution stabilizes. Once a fixed-K solution becomes sufficiently mature, adaptive path refinement evaluates alternative switch penalties to improve predictive performance and path compactness. Because decoding uses observed next-step targets and sequence-level evidence, it is restricted to offline discovery and annotation.

Adaptive regime-space construction. For a given candidate regime-space size K, the inner procedure optimizes regime assignments and predictive dynamics but does not assess whether the structure is appropriate. RegimeUnfold therefore adapts the regime space through split, merge, and same-K refinement. Split proposals examine regimes with heterogeneous predictive dynamics from multiple complementary views to generate candidate binary partitions. Merge proposals identify potentially redundant regimes primarily through predictive substitutability, supplemented by similarities in prediction, prototype, and bias, together with temporal adjacency. Same-K refinement reorganizes the partition without changing the regime number, either by removing an unsupported regime introduced by a provisional split or by splitting one regime and merging a different pair. All resulting candidates are then trained and re-decoded before entering a unified acceptance procedure.

For each candidate, the predictive performance score P_s and regime-path quality score T_s of sequence s are combined as $J_s = \lambda_P \,\text{Norm}_{\text{str}}(P_s) + \lambda_T \,\text{Norm}_{\text{str}}(T_s)$, where $\text{Norm}_{\text{str}}(\cdot)$ rescales the two terms and λ_P and λ_T control their relative contributions. To reduce bias from incomplete regime coverage, the training and validation scores are further combined as $J(\mathcal{M}) =$

$[\omega_{\mathrm{tr}} J_{\mathrm{tr}}(\mathcal{M}) + \omega_{\mathrm{val}} J_{\mathrm{val}}(\mathcal{M})]/(\omega_{\mathrm{tr}} + \omega_{\mathrm{val}})$, where $\mathcal{M}$ denotes the candidate model and $\omega_{\mathrm{val}} > \omega_{\mathrm{tr}}$. Since a smaller $J(\mathcal{M})$ indicates a better candidate, the global-score criterion is satisfied only when the proposed structural change yields a sufficient improvement over the pre-change structure. Structural reliability further requires every active regime to have sufficient effective usage and a sufficient number of valid persistent segments. Finally, voting over fixed local regime segments extracted from the mature pre-change path checks whether the structural change is consistently supported across the sequence, and passes when the supporting votes reach the required operation-specific threshold. A structural change is accepted only when all three criteria are satisfied.

The structural search terminates when no split, merge, or same-K refinement is accepted in a complete cycle or when the maximum allowed regime-space size is reached. Accepted candidates across different values of K are then compared. A prediction guard removes candidates whose validation predictive performance is noticeably worse than that of a simpler accepted model. Among the remaining candidates, RegimeUnfold retains models whose global scores lie within a predefined tolerance of the lowest score and selects the smallest K, avoiding unnecessary increases in regime-space complexity for marginal improvement. The resulting model defines the final offline regime space and provides the regime-conditioned predictors and regime annotations for online forecasting.

Detailed definitions of the offline loss terms and procedures for local regime scoring, path refinement, structural proposal generation, candidate evaluation, voting, and cross-K selection are provided in Appendix A.

4.2 Online Regime Evolution

The offline regime decoder relies on observed next-step targets and sequence-level evidence, which are unavailable during real-time forecasting. The online stage therefore performs causal regime-aware forecasting using only information available up to the current time. RegimeUnfold selects the forecasting regime from the current regime space and, as new observations become available, continually assesses whether the existing regime space remains adequate for the evolving predictive dynamics. When persistent evidence indicates that the current regime space is insufficient, the model determines whether the change can be accommodated by updating an existing regime or requires the creation of a new regime under stability constraints. Accordingly, online regime evolution consists of regime selection, regime-space evolution assessment, and stability-aware adaptation.

Regime selection and forecasting within the current regime space. RegimeUnfold constructs an online regime selection router whose parameters are shared across regimes in the current regime space $\mathcal{R}_t$, allowing the same routing mechanism to operate as the regime space changes over time without requiring a fixed number of regimes. For each candidate regime $k \in \mathcal{R}_t$, predictive and temporal evidence are evaluated separately. A shared context encoder summarizes recent observations, latent representations, and shared-base prediction residuals, while a shared regime-history encoder captures the recent predictive behavior and prototype compatibility of regime k. Together, they produce a normalized predictive score $\mathrm{Score}^{(k)}_{\mathrm{pred},t}$. In parallel, a shared regime-identity encoder represents each regime, and the temporal branch evaluates its compatibility with the previously inferred regime and its duration to obtain a normalized temporal score $\mathrm{Score}^{(k)}_{\mathrm{temp},t}$.

The two scores are adaptively fused as

$$\mathrm{Score}^{(k)}_t = (1-\alpha_t)\mathrm{Score}^{(k)}_{\mathrm{pred},t} + \alpha_t \mathrm{Score}^{(k)}_{\mathrm{temp},t}, \qquad \alpha_t \in [0,1], \tag{3}$$

where α_t is learned from the current routing context together with the confidence and agreement of the predictive and temporal branches, adaptively controlling their relative contributions. RegimeUnfold then performs hard regime decoding by maximum fused score,

$$\widehat{z}_{t+1} = \arg\max_{k \in \mathcal{R}_t} \mathrm{Score}^{(k)}_t, \qquad \widehat{\mathbf{x}}_{t+1} = \widehat{\mathbf{x}}^{(\widehat{z}_{t+1})}_{t+1}, \tag{4}$$

so that only the predictor associated with the selected regime contributes to the forecast. The router is trained using the offline regime annotations as routing supervision, with regime-specific prediction costs providing additional forecasting guidance.

Evidence-driven regime-space evolution assessment. Regime selection assumes that the incoming dynamics can still be represented by one of the regimes in the current regime space. After

each new observation becomes available, RegimeUnfold therefore evaluates whether this assumption remains valid. The regime evolution router compares the observed dynamics with the predictive and structural references of all regimes in $\mathcal{R}_t$. The resulting regime-wise predictive and structural discrepancies are aggregated into an evidence representation $\mathbf{v}_t$, from which two complementary evidence scores are produced $[\mathrm{Score}_{\mathrm{ins},t}, \mathrm{Score}_{\mathrm{nov},t}] = \sigma(G_\eta(\mathbf{v}_t))$, where G_η denotes the evolution-scoring network parameterized by η and $\sigma(\cdot)$ denotes the elementwise sigmoid function. $\mathrm{Score}_{\mathrm{ins},t}$ measures predictive insufficiency, indicating whether the existing regime predictors can still adequately explain the current observation relative to their historical predictive behavior, while $\mathrm{Score}_{\mathrm{nov},t}$ measures structural novelty, indicating whether the observed dynamics depart from the structural references of the current regime space. The two evidence scores are not mutually exclusive and may be simultaneously high.

The two evidence scores are learned with complementary supervision. Predictive-insufficiency supervision is constructed from excessive prediction errors relative to regime-specific historical references, whereas structural-novelty supervision is generated by selectively hiding annotated regimes from the available regime set and excluding information associated with the hidden regime. During online operation, the two scores are assessed using independently calibrated thresholds. Predictive-insufficiency evidence is accumulated when it is supported by sufficiently large realized prediction degradation, while structural-novelty evidence must persist across consecutive observations. If the accumulated evidence remains insufficient, forecasting continues within the current regime space; once sufficiently supported, the evidence activates stability-aware regime-space adaptation.

Stability-aware regime-space adaptation. Persistent evolution evidence does not immediately modify the deployed model, since premature adaptation may disrupt established regime-specific dynamics or unnecessarily expand the regime space. Instead, RegimeUnfold constructs UPDATE and CREATE alternatives from the same accumulated adaptation episode. UPDATE tests whether the observed change can be absorbed as drift within an existing regime by individually adapting a small set of plausible regimes and retaining the strongest validated adaptation. CREATE instead tests whether the change requires a new regime by training an additional regime-specific predictor while preserving the established regimes.

The UPDATE and CREATE candidates are evaluated together with the current model under a common protocol. Predictive performance requires the adaptation benefit to be supported by held-out recent observations, while stability preservation requires established-regime forecasting and regime assignments to remain sufficiently stable. CREATE is subject to an additional validity requirement, requiring persistent support and sufficient distinction from existing regimes. If UPDATE is selected, only the validated regime-specific predictor is replaced and the regime space remains unchanged; if CREATE is selected, the new predictor and its regime-specific references are incorporated into the regime space. Once sufficient validation evidence is available, the comparison yields one of three outcomes: UPDATE an existing regime, CREATE a new regime, or KEEP CURRENT when neither adaptation provides sufficiently supported improvement over the current model, in which case additional evidence continues to accumulate. Until the decision is resolved, a lightweight provisional predictor temporarily supports forecasting without being incorporated into the established regime space and is removed once the decision is finalized.

Detailed formulations and procedures for regime selection, evolution-evidence assessment, and stability-aware regime-space adaptation are provided in Appendix B.

5 Experiments

We extensively evaluate RegimeUnfold on multivariate sequential datasets with latent regime changes to assess both forecasting accuracy and regime inference performance.

Five synthetic datasets and four real-world datasets are used for evaluation. The synthetic datasets are generated from higher-order semi-Markov regime processes with state-dependent autoregressive emissions. Four datasets maintain a fixed regime space with different regime numbers and transition frequencies, while the Progressive dataset progressively expands the available regime set to evaluate regime-space evolution as previously unseen predictive dynamics emerge. Detailed data-generating processes and parameter settings are provided in Appendix C. The real-world datasets include the Server Machine Dataset (SMD), an exchange-rate dataset (Exchange), a traffic-speed

Table 1: Main results on synthetic datasets.

Model	Fixed-K2				Fixed-K3				Progressive			
	Accuracy (%)	F1 (%)	MAE	MSE	Accuracy (%)	F1 (%)	MAE	MSE	Accuracy (%)	F1 (%)	MAE	MSE
HMM	83.59	82.42	0.0735	0.01289	72.10	71.53	0.0395	0.00497	51.81	35.31	0.0590	0.00577
HSMM	54.19	51.45	0.0569	0.00635	61.11	60.62	0.0385	0.00413	51.21	34.81	0.0602	0.00601
HOHMM	82.24	81.12	0.0737	0.01296	76.77	76.07	0.0429	0.00591	51.81	35.31	0.0590	0.00577
BR-iHMM	96.58	95.92	0.0094	0.00028	73.64	59.07	0.0101	0.00041	45.53	30.41	0.0215	0.00167
NHMM	70.45	41.33	0.0224	0.00127	44.56	20.55	0.0206	0.00105	30.47	11.68	0.0237	0.00141
NCTRL	81.15	69.34	0.0212	0.00144	63.08	55.36	0.0280	0.00197	41.25	26.47	0.0342	0.00284
Markovian-RNN	70.08	42.74	0.0103	0.00023	44.61	20.98	0.0126	0.00037	30.88	13.09	0.0171	0.00096
DEN-HMM	71.55	47.72	0.0707	0.01084	46.27	32.58	0.0392	0.00472	66.27	52.52	0.0471	0.00419
DRL-STAF	93.99	93.11	0.0099	0.00029	66.31	58.05	0.0160	0.00086	58.60	54.81	0.0330	0.00278
iTransformer	-	-	0.0248	0.00128	-	-	0.0205	0.00105	-	-	0.0280	0.00176
PatchTST	-	-	0.0241	0.00128	-	-	0.0205	0.00105	-	-	0.0214	0.00123
TimeMixer	-	-	0.0278	0.00173	-	-	0.0242	0.00154	-	-	0.0401	0.00367
RegimeUnfold	**97.85**	**97.68**	**0.0093**	**0.00022**	**96.31**	**95.88**	**0.0093**	**0.00026**	**91.49**	**91.19**	**0.0134**	**0.00081**

Table 2: Main results on real-world datasets.

Model	SMD				Exchange		Traffic		GOLD	
	Accuracy (%)	F1 (%)	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
HMM	54.16	48.42	0.1341	0.02883	7.9617	248.7797	5.4173	67.6396	984.65	1381161.90
HSMM	55.60	50.25	0.1327	0.02831	7.9634	248.8891	5.4124	67.3115	984.78	1381436.92
HOHMM	54.81	49.07	0.1320	0.02792	7.9624	248.8203	5.4171	67.6296	984.64	1381160.69
BR-iHMM	70.66	41.40	**0.0183**	0.00096	**0.2119**	0.2570	1.5522	**5.5274**	12.70	392.96
NHMM	70.66	41.40	0.0221	0.00109	1.8489	17.4428	1.5881	6.7800	250.44	248913.75
NCTRL	80.72	73.65	0.0308	0.00257	3.5346	61.5884	2.1324	10.9013	290.13	284846.18
Markovian-RNN	70.66	42.20	0.0185	0.00095	0.2642	0.3730	1.5429	5.8620	16.10	631.43
DEN-HMM	70.80	42.09	0.1501	0.03730	10.8710	427.8633	7.6843	107.7766	1206.41	1891706.24
DRL-STAF	85.63	81.60	0.0218	0.00149	2.1519	19.6272	1.5406	6.6540	314.72	233775.77
iTransformer	-	-	0.0241	0.00127	0.6834	2.9078	1.6371	5.9855	275.24	361721.66
PatchTST	-	-	0.0212	0.00103	0.3539	0.6805	1.6144	5.8363	181.02	128705.79
TimeMixer	-	-	0.0261	0.00183	3.8829	69.0204	1.8315	7.6385	499.04	928360.28
RegimeUnfold	**89.65**	**85.82**	**0.0183**	**0.00092**	0.2146	**0.2561**	**1.5323**	5.7416	**11.82**	**295.19**

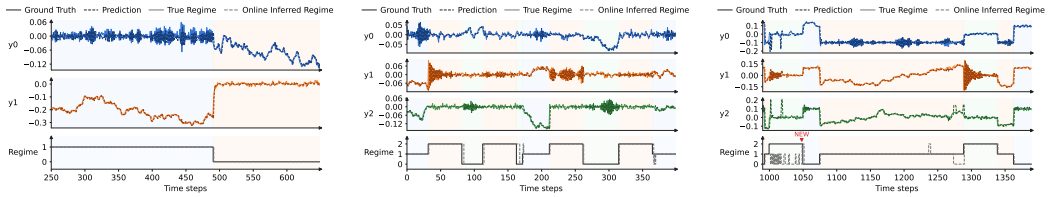
dataset (Traffic), and a gold futures dataset (GOLD), with detailed descriptions and preprocessing settings provided in Appendix D.

We choose twelve representative models as baselines, including (1) classical and nonparametric HMM-based models: HMM, HSMM, HOHMM, and BR-iHMM (Yiu et al., 2026); (2) DL-HMM hybrids: NHMM (Tran et al., 2016), NCTRL (Song et al., 2023), Markovian-RNN (Ilhan et al., 2023), DEN-HMM (Bansal & Zhou, 2025), and DRL-STAF (Jiang et al., 2026); and (3) time-series forecasting models: iTransformer (Liu et al., 2024), PatchTST (Nie et al., 2023), and TimeMixer (Wang et al., 2024).

5.1 Main Results

RegimeUnfold is evaluated from two aspects: forecasting performance and regime inference performance. Forecasting performance is measured using mean absolute error (MAE) and mean squared error (MSE), while regime inference performance is evaluated using accuracy and F1 score. Lower MAE and MSE indicate better forecasting performance, whereas higher accuracy and F1 indicate better regime inference. The best results are highlighted in **red**, and the second-best results are underlined. The complete results can be found in Appendix E.

Results on synthetic datasets. Table 1 reports the results on three representative synthetic datasets. RegimeUnfold achieves the best overall performance in both regime inference and forecasting across the fixed-regime and progressive settings. On Fixed-$K2$ and Fixed-$K3$, the high Accuracy and F1 scores together with low forecasting errors show that the learned regime space can effectively separate distinct predictive dynamics and support accurate regime-conditioned forecasting. The advantage becomes more pronounced on the Progressive dataset, where RegimeUnfold substantially improves regime inference while maintaining the lowest forecasting errors as the regime space evolves. Figure 2 further presents the forecasting and regime inference results on these three synthetic datasets. The different background colors denote the ground-truth regimes, while the bottom panels compare the true and online inferred regime sequences. RegimeUnfold accurately tracks the observations and recovers regime transitions on the fixed-regime datasets. On the Progressive dataset, the emerging regime is incorporated after a period of evidence accumulation, resulting in a delay in the "NEW" decision at its first appearance. Once incorporated, however, the new regime is correctly reused when the same dynamics reappear later, enabling stable regime



(a) Fixed-K2 dataset. (b) Fixed-K3 dataset. (c) Progressive dataset.

Figure 2: Representative forecasting and regime inference results on the synthetic datasets.

Table 3: Ablation results on synthetic datasets.

Model	Fixed-K2				Fixed-K3				Progressive			
	Accuracy (%)	F1 (%)	MAE	MSE	Accuracy (%)	F1 (%)	MAE	MSE	Accuracy (%)	F1 (%)	MAE	MSE
RegimeUnfold w/o Regime	-	-	0.0236	0.00131	-	-	0.0186	0.00095	-	-	0.0304	0.00229
RegimeUnfold w/o Evolution	97.75	97.58	0.0100	0.00034	**96.33**	**95.90**	0.0094	0.00026	51.56	34.08	0.0426	0.00508
RegimeUnfold	**97.85**	**97.68**	**0.0093**	**0.00022**	96.31	95.88	**0.0093**	**0.00026**	**91.49**	**91.19**	**0.0134**	**0.00081**

tracking and forecasting. These results demonstrate that RegimeUnfold can model regime-specific predictive dynamics while adapting the regime space when previously unseen dynamics emerge.

Results on real-world datasets. The SMD dataset provides ground-truth anomaly labels, allowing us to evaluate both forecasting and regime inference, whereas only forecasting is evaluated on Exchange, Traffic, and GOLD. As shown in Table 2, RegimeUnfold achieves the best Accuracy and F1 on SMD together with the lowest MSE. On the datasets without regime annotations, it achieves the lowest MSE on Exchange, the lowest MAE on Traffic, and the lowest MAE and MSE on GOLD, while remaining competitive on the other metrics. These results suggest that the learned latent regime structure captures predictive heterogeneity that remains useful even without explicit regime annotations, and that the benefits of RegimeUnfold extend to diverse real-world sequential data.

5.2 Ablation Study

To examine the contributions of explicit regime modeling and regime-space evolution, we compare RegimeUnfold with RegimeUnfold w/o Regime and RegimeUnfold w/o Evolution. Removing regime modeling substantially increases forecasting errors across all three synthetic datasets, demonstrating the benefit of explicitly modeling heterogeneous regime-specific predictive dynamics. On the fixed-regime datasets, RegimeUnfold and RegimeUnfold w/o Evolution achieve very similar regime inference performance, while the full model attains lower forecasting errors overall. In contrast, on the Progressive dataset, disabling regime-space evolution causes a substantial deterioration in both regime inference and forecasting performance. These results show that explicit regime modeling improves forecasting under heterogeneous dynamics, while regime-space evolution becomes particularly important when previously unseen dynamics emerge.

6 Conclusion

This paper proposes RegimeUnfold, a regime-aware forecasting framework with an evolving latent regime space. RegimeUnfold discovers and models latent regimes offline without requiring their number to be specified in advance, and continually adapts the learned regime space online through hard regime decoding, evolution assessment, and stability-aware regime adaptation. Experiments on five synthetic and four real-world datasets demonstrate strong forecasting performance across diverse settings and reliable regime inference where reference labels are available. The results further show that explicit regime modeling improves the representation of heterogeneous predictive dynamics, while regime-space evolution enables the model to accommodate drifting and previously unseen dynamics during online forecasting. Future work will focus on reducing the delay in detecting and incorporating emerging regimes while preserving conservative stability control, and on exploring the integration of time-series foundation models to improve representation transfer and adaptation across heterogeneous sequential domains.

AI USE STATEMENT

In this work, generative AI tools were used to assist with language editing and polishing of the manuscript, including improvements to grammar, wording, and readability. They were also used for retrieval and discovery to help check whether any published studies relevant to this work had been overlooked during the literature review. In addition, generative AI tools assisted with code debugging and review, including identifying and removing implementation bugs, checking whether the implemented logic was consistent with the planned methodology, and improving code organization and readability. All retrieved literature, code-related suggestions, and manuscript revisions were independently reviewed and verified by the authors before use. The authors take full responsibility for the final research content, implementation, and manuscript.

ETHICS STATEMENT

This work adheres to the ICLR Code of Ethics. The study does not involve human subjects, sensitive personal information, or applications with immediate potential for harm. The experiments are conducted on synthetic or research datasets and do not involve interventions or decisions affecting individuals. We are not aware of any conflicts of interest, sponsorship concerns, or other ethical risks associated with this research. All research procedures, experimental analyses, and reported results were conducted and reviewed in accordance with standard principles of research integrity.

REPRODUCIBILITY STATEMENT

We have made extensive efforts to support reproducibility. The proposed RegimeUnfold framework and its main algorithmic procedures are described in the main text, with offline regime discovery and online regime evolution detailed in Sections 4.1 and 4.2, respectively. Complete formulations and procedures for the two stages are provided in Appendices A and B, including the structural-search procedures for offline discovery and the detailed routing, evolution-assessment, and adaptation procedures for online evolution. The datasets, preprocessing procedures, evaluation metrics, baseline models, and experimental results are described in Section 5 and Appendices C-E. The synthetic datasets are generated using explicitly specified data-generating processes and parameters, while the real-world datasets are accompanied by their sources and preprocessing procedures. Source code, configuration files, and scripts used to reproduce the experiments will be made available at `https://anonymous.4open.science/r/RegimeUnfold`.

REFERENCES

Vipul Bansal and Shiyu Zhou. DEN-HMM: Deep emission network based hidden Markov model with time-evolving multivariate observations. *IISE Transactions*, 57(12):1450–1463, 2025.

Matthew J. Beal, Zoubin Ghahramani, and Carl E. Rasmussen. The infinite hidden Markov model. In *NeurIPS*, volume 14, pp. 577–584, 2001.

Defu Cao, Yujing Wang, Juanyong Duan, Ce Zhang, Xia Zhu, Congrui Huang, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, and Qi Zhang. Spectral temporal graph neural network for multivariate time-series forecasting. In *NeurIPS*, volume 33, pp. 17766–17778, 2020.

Sunrit Chakraborty, Aritra Guha, Rayleigh Lei, and XuanLong Nguyen. Scalable nonparametric bayesian learning for dynamic velocity fields. In *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, volume 216 of *Proceedings of Machine Learning Research*, pp. 282–292. PMLR, 2023.

G. E. Dahl, Dong Yu, Li Deng, and A. Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):30–42, 2012.

Colin Reimer Dawson, Chaofan Huang, and Clayton T. Morrison. An infinite hidden Markov model with similarity-biased transitions. In *ICML*, volume 70, pp. 942–950, 2017.

Emily B. Fox, Erik B. Sudderth, Michael I. Jordan, and Alan S. Willsky. An HDP-HMM for systems with state persistence. In *ICML*, pp. 312–319, 2008a.

Emily B. Fox, Erik B. Sudderth, Michael I. Jordan, and Alan S. Willsky. Nonparametric bayesian learning of switching linear dynamical systems. In *NeurIPS*, volume 21, pp. 457–464, 2008b.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *COLM*, 2024.

Michael C. Hughes, William T. Stephenson, and Erik B. Sudderth. Scalable adaptation of state complexity for nonparametric hidden Markov models. In *NeurIPS*, volume 28, pp. 1198–1206, 2015.

Fatih Ilhan, Oguzhan Karaahmetoglu, Ismail Balaban, and Suleyman Serdar Kozat. Markovian RNN: An Adaptive Time Series Prediction Network With HMM-Based Switching for Nonstationary Environments. *IEEE Transactions on Neural Networks and Learning Systems*, 34(2): 715–728, 2023.

Manrui Jiang, Jingru Huang, Yong Chen, and Chen Zhang. DRL-STAF: A deep reinforcement learning framework for state-aware forecasting of complex multivariate hidden markov processes. In *ICML*, 2026.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long- and short-term temporal patterns with deep neural networks. In *ACM SIGIR*, pp. 95–104, 2018.

Wanshan Li and Chen Zhang. A Markov-switching hidden heterogeneous network autoregressive model for multivariate time series data with multimodality. *IISE Transactions*, 55(11):1118–1132, 2023.

Chiu-Hsiang Lin, Kung-Jeng Wang, Ahmed Abide Tadesse, and Bereket Haile Woldegiorgis. Human-robot collaboration empowered by hidden semi-Markov model for operator behaviour prediction in a smart assembly system. *Journal of Manufacturing Systems*, 62:317–333, 2022.

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

Sanae Lotfi, Pavel Izmailov, Gregory Benton, Micah Goldblum, and Andrew Gordon Wilson. Bayesian Model Selection, the Marginal Likelihood, and Generalization. In *ICML*, pp. 14223–14247, 2022.

Bhavya Mor, Sunita Garhwal, and Ajay Kumar. A systematic review of hidden markov models and their applications. *Archives of Computational Methods in Engineering*, 28(3):1429–1448, 2021.

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

Victor Rodriguez-Fernandez, Antonio Gonzalez-Pardo, and David Camacho. Modelling behaviour in UAV operations using higher order double chain markov models. *IEEE Computational Intelligence Magazine*, 12(4):28–37, 2017.

Ardavan Saeedi, Matthew D. Hoffman, Matthew J. Johnson, and Ryan P. Adams. The segmented iHMM: A simple, efficient hierarchical infinite HMM. In *ICML*, volume 48, pp. 2682–2691, 2016.

Xiangchen Song, Weiran Yao, Yewen Fan, Xinshuai Dong, Guangyi Chen, Juan Carlos Niebles, Eric Xing, and Kun Zhang. Temporally disentangled representation learning under unknown nonstationarity. In *NeurIPS*, volume 36, pp. 8092–8113, 2023.

Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In *SIGKDD*, pp. 2828–2837, 2019.

K. M. Tran, Y. Bisk, A. Vaswani, D. Marcu, and K. Knight. Unsupervised neural hidden markov models. In *Proceedings of the Workshop on Structured Prediction for NLP*, pp. 63–71, 2016.

Nilesh Tripuraneni, Shixiang Gu, Hong Ge, and Zoubin Ghahramani. Particle gibbs for infinite hidden markov models. In *NeurIPS*, volume 28, pp. 2395–2403, 2015.

Jurgen Van Gael, Yunus Saatci, Yee Whye Teh, and Zoubin Ghahramani. Beam sampling for the infinite hidden Markov model. In *ICML*, pp. 1088–1095, 2008.

Senzhang Wang, Xiaoming Zhang, Fengxiang Li, Philip S. Yu, and Zhiqiu Huang. Efficient traffic estimation with multi-sourced data by parallel coupled hidden markov model. *IEEE Transactions on Intelligent Transportation Systems*, 20(8):3010–3023, 2019.

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 *ICLR*, 2024.

Horace Yiu, Leandro Sánchez-Betancourt, Álvaro Cartea, and Gerardo Duran-Martin. Doubly outlier-robust online infinite hidden Markov model. In *ICML*, 2026.

Aonan Zhang, Jun Zhu, and Bo Zhang. Max-margin infinite hidden Markov models. In *ICML*, volume 32, pp. 315–323, 2014.

A Details of Offline Regime Discovery

This section provides the detailed formulations and procedures underlying the three main parts of offline regime discovery: regime-conditioned dynamics modeling, offline regime inference and joint learning, and adaptive regime-space construction.

A.1 Regime-Conditioned Dynamics Model

This subsection provides the detailed parameterization and training objective of the regime-conditioned dynamics model introduced in Section 4.1. Throughout this subsection, N denotes the number of valid one-step training pairs, d the output dimension, and K the number of active regimes.

A.1.1 Model parameterization

Given the observation history $\mathbf{H}_t$, the deep encoder produces

$$\mathbf{h}_t = E_\theta(\mathbf{H}_t), \qquad \mathbf{c}_t = Q_\theta(\mathbf{h}_t), \tag{5}$$

where $\mathbf{h}_t$ is the shared representation and $\mathbf{c}_t$ is the low-dimensional regime code. Each active regime k is associated with a residual predictor r_k, a bias $\mathbf{b}_k$, and a prototype $\mathbf{p}_k$. The prototype compatibility used for regime inference and structural comparison is measured by

$$d_{t,k} = \|\mathbf{c}_t - \mathbf{p}_k\|_2^2 . \tag{6}$$

The compact masked representation $\widetilde{\mathbf{h}}_t^{(k)}$ used in the main text summarizes a guarded boundary-masking mechanism that selectively suppresses history preceding a reliable regime transition. For boundary detection, let $\mathbf{h}_{t,\ell}$ denote the encoder representation at history position $\ell \in \{1, \dots, h\}$ and let $\mathbf{c}_{t,\ell} = Q_\theta(\mathbf{h}_{t,\ell})$ be its corresponding regime code. For each active regime k, the token-level prototype distance is $d^{\mathrm{tok}}_{t,\ell,k} = \|\mathbf{c}_{t,\ell} - \mathbf{p}_k\|_2^2$, and the closest competing-regime distance is $d^{\mathrm{tok}}_{t,\ell,-k} = \min_{j \neq k} d^{\mathrm{tok}}_{t,\ell,j}$. Their relative difference defines the preference of history position ℓ for regime k:

$$u_{t,\ell,k} = \frac{d^{\mathrm{tok}}_{t,\ell,-k} - d^{\mathrm{tok}}_{t,\ell,k}}{d^{\mathrm{tok}}_{t,\ell,-k} + d^{\mathrm{tok}}_{t,\ell,k} + \epsilon}, \tag{7}$$

where $\epsilon > 0$ is a small constant for numerical stability. A positive $u_{t,\ell,k}$ indicates that the history position is closer to regime k than to its closest competitor, while a negative value indicates the opposite.

For each possible boundary position $r \in \{1, \dots, h-1\}$, the average regime preferences before and after the boundary are

$$U^{\mathrm{pre}}_{t,k}(r) = \frac{1}{r} \sum_{\ell=1}^{r} u_{t,\ell,k}, \qquad U^{\mathrm{post}}_{t,k}(r) = \frac{1}{h-r} \sum_{\ell=r+1}^{h} u_{t,\ell,k}. \tag{8}$$

A candidate boundary is considered reliable only when the preference changes from another regime before the boundary to regime k after the boundary. Accordingly, its boundary score is

$$s_{t,k}(r) = \min\left\{ \left[\frac{U_{t,k}^{\mathrm{post}}(r) - U_{t,k}^{\mathrm{pre}}(r)}{2} \right]_{[0,1]}, \left[U_{t,k}^{\mathrm{post}}(r) \right]_{[0,1]}, \left[-U_{t,k}^{\mathrm{pre}}(r) \right]_{[0,1]} \right\}, \tag{9}$$

where $[x]_{[0,1]} = \min(1, \max(0, x))$. The selected boundary and its evidence strength are

$$r_{t,k}^{*} = \arg\max_{r} s_{t,k}(r), \qquad s_{t,k}^{*} = \max_{r} s_{t,k}(r). \tag{10}$$

Each regime has a learnable boundary margin $\mu_k \in (0, 1)$. The corresponding activation gate is defined as

$$a_{t,k} = \left[\frac{s_{t,k}^{*} - \mu_k}{1 - \mu_k} \right]_{[0,1]}. \tag{11}$$

Boundary evidence below μ_k leaves the residual branch unchanged, while stronger evidence gradually increases the contribution of the masked-history residual.

When $a_{t,k} > 0$, a raw temporal visibility mask is constructed as

$$M_{t,\ell}^{(k)} = \begin{cases} \eta_{\mathrm{mask}}, & \ell \le r_{t,k}^{*}, \\ 1, & \ell > r_{t,k}^{*}, \end{cases} \tag{12}$$

where $\eta_{\mathrm{mask}} \in (0, 1)$ is a small visibility floor. Observations preceding the detected boundary are therefore downweighted rather than completely removed. Re-encoding the history under $M_{t,\ell}^{(k)}$ produces the masked-history representation $\mathbf{h}_{t,\mathrm{mask}}^{(k)}$.

The residual predictor is evaluated along both the full-history and masked-history paths:

$$\boldsymbol{\rho}_{t,k}^{\mathrm{full}} = r_k(\mathbf{h}_t), \qquad \boldsymbol{\rho}_{t,k}^{\mathrm{mask}} = r_k(\mathbf{h}_{t,\mathrm{mask}}^{(k)}). \tag{13}$$

The final regime-specific residual is obtained by guarded interpolation:

$$\boldsymbol{\rho}_{t,k} = (1 - a_{t,k})\boldsymbol{\rho}_{t,k}^{\mathrm{full}} + a_{t,k}\boldsymbol{\rho}_{t,k}^{\mathrm{mask}}. \tag{14}$$

Consequently, the regime-conditioned prediction is

$$\widehat{\mathbf{x}}_{t+1}^{(k)} = f_0(\mathbf{h}_t) + \boldsymbol{\rho}_{t,k} + \mathbf{b}_k. \tag{15}$$

When no reliable boundary is detected, $a_{t,k} = 0$ and the model retains the full-history residual path. As the boundary evidence strengthens, $a_{t,k}$ approaches one and the prediction increasingly relies on the masked-history residual. The discrete boundary selection is treated as non-differentiable, while the regime-specific boundary margin remains learnable through the gate in Eq. (11).

A.1.2 Offline training objective

Given hard regime assignments, the regime-conditioned dynamics model is trained with the composite objective

$$\begin{aligned} \mathcal{L}_{\mathrm{dyn}} = & \lambda_{\mathrm{pred}}\mathcal{L}_{\mathrm{pred}} + \lambda_{\mathrm{base}}\mathcal{L}_{\mathrm{base}} + \lambda_{\mathrm{com}}\mathcal{L}_{\mathrm{com}} + \lambda_{\mathrm{sep}}\mathcal{L}_{\mathrm{sep}} + \lambda_{\mathrm{spec}}\mathcal{L}_{\mathrm{spec}} \\ & + \lambda_{\mathrm{res}}\mathcal{L}_{\mathrm{res}} + \lambda_{\mathrm{bias}}\mathcal{L}_{\mathrm{bias}} + \lambda_{\mathrm{off}}\mathcal{L}_{\mathrm{off}} + \lambda_{\mathrm{mask}}\mathcal{L}_{\mathrm{mask}}. \end{aligned} \tag{16}$$

The individual terms follow the same functional grouping as in the main text and are detailed below.

Prediction losses. The assigned-regime prediction loss is

$$\mathcal{L}_{\mathrm{pred}} = \frac{1}{Nd} \sum_{t} \left\| \mathbf{x}_{t+1} - \widehat{\mathbf{x}}_{t+1}^{(z_{t+1})} \right\|_2^2. \tag{17}$$

This term directly trains the predictor associated with the assigned regime.

The shared-base predictor is trained using

$$\mathcal{L}_{\mathrm{base}} = \frac{1}{Nd} \sum_{t} \left\| \mathbf{x}_{t+1} - f_0(\mathbf{h}_t) \right\|_2^2. \tag{18}$$

Maintaining an informative shared predictor encourages the regime-specific residual branches to focus on deviations from common predictive dynamics rather than relearning the full prediction function independently.

Regime representation and specialization losses. The prototype-commitment loss encourages each regime code to remain close to the prototype of its assigned regime:

$$\mathcal{L}_{\text{com}} = \frac{1}{N}\sum_{t} d_{t,z_{t+1}} = \frac{1}{N}\sum_{t} \left\|\mathbf{c}_t - \mathbf{p}_{z_{t+1}}\right\|_2^2. \tag{19}$$

To maintain separation among different regime prototypes, RegimeUnfold further uses

$$\mathcal{L}_{\text{sep}} = \frac{1}{K(K-1)}\sum_{k\neq j}\left[\delta_p - \|\mathbf{p}_k - \mathbf{p}_j\|_2\right]_+^2, \tag{20}$$

where δ_p is the prototype-separation margin and $[u]_+ = \max(u, 0)$.

Regime specialization is encouraged by comparing the assigned predictor with competing regime predictors. Let

$$e_{t,k} = \frac{1}{d}\left\|\mathbf{x}_{t+1} - \widehat{\mathbf{x}}_{t+1}^{(k)}\right\|_2^2 \tag{21}$$

denote the prediction error of regime k, and define the adaptive margin as

$$m = \alpha_{\text{spec}}\,\text{median}_{t,k}(e_{t,k}), \tag{22}$$

where α_{spec} controls the margin scale. The specialization loss is

$$\mathcal{L}_{\text{spec}} = \frac{1}{N}\sum_{t}\left[e_{t,z_{t+1}} - \text{sg}\left(\min_{j\neq z_{t+1}} e_{t,j}\right) + m\right]_+, \tag{23}$$

where $\text{sg}(\cdot)$ denotes the stop-gradient operator. This term encourages the assigned regime predictor to outperform the strongest competing predictor by an adaptive margin.

Regime-specific regularization. The magnitude of the selected residual output is regularized by

$$\mathcal{L}_{\text{res}} = \frac{1}{Nd}\sum_{t}\left\|\boldsymbol{\rho}_{t,z_{t+1}}\right\|_2^2, \tag{24}$$

while the regime-specific bias is controlled through

$$\mathcal{L}_{\text{bias}} = \frac{1}{Nd}\sum_{t}\left\|\mathbf{b}_{z_{t+1}}\right\|_2^2. \tag{25}$$

To prevent non-assigned residual predictors from developing generic correction functions, RegimeUnfold additionally uses

$$\mathcal{L}_{\text{off}} = \frac{1}{Nd(K-1)}\sum_{t}\sum_{k\neq z_{t+1}}\|\boldsymbol{\rho}_{t,k}\|_2^2. \tag{26}$$

Temporal-mask regularization. The activation of the guarded temporal mask is regularized by

$$\mathcal{L}_{\text{mask}} = \frac{1}{N}\sum_{t} a_{t,z_{t+1}}. \tag{27}$$

This term discourages unnecessary activation of the guarded temporal mask and favors the full-history residual path unless sufficient boundary evidence supports temporal masking.

A.2 Offline Regime Inference and Joint Learning

This subsection provides the detailed formulation of offline hard regime inference and its alternating optimization with the regime-conditioned dynamics model.

A.2.1 LOCAL ASSIGNMENT EVIDENCE

Given the current dynamics model, RegimeUnfold first constructs a local assignment cost for each sample-regime pair. For output dimension q, the squared prediction error of regime k is

$$e_{t,k,q} = \left(x_{t+1,q} - \widehat{x}^{(k)}_{t+1,q}\right)^2. \tag{28}$$

To account for both overall predictive fit and large discrepancies in individual outputs, the output-wise errors are summarized as

$$\bar{e}_{t,k} = \omega_{\text{mean}} \frac{1}{d} \sum_{q=1}^{d} e_{t,k,q} + \omega_{\text{worst}} \max_{1 \le q \le d} e_{t,k,q}, \tag{29}$$

where $\omega_{\text{mean}}, \omega_{\text{worst}} \ge 0$ and $\omega_{\text{mean}} + \omega_{\text{worst}} = 1$. The first term measures the average predictive discrepancy across outputs, while the second prevents a large error in a single output from being concealed by accurate predictions on the remaining dimensions.

Because squared prediction errors discard the direction of the deviation, RegimeUnfold additionally computes the signed prediction bias

$$\xi_{t,k} = \frac{1}{d} \sum_{q=1}^{d} \left(x_{t+1,q} - \widehat{x}^{(k)}_{t+1,q}\right). \tag{30}$$

The sign of $\xi_{t,k}$ indicates the direction of the prediction bias, with positive and negative values corresponding to underprediction and overprediction, respectively, while its squared magnitude $\xi^2_{t,k}$ measures the strength of this directional bias for incorporation into the nonnegative assignment cost.

Prediction-based evidence may remain ambiguous when multiple regimes produce similar one-step forecasts. RegimeUnfold therefore also incorporates the prototype distance $d_{t,k}$ defined in the preceding subsection as complementary representation-level evidence.

The predictive error $\bar{e}_{t,k}$, squared directional-bias evidence $\xi^2_{t,k}$, and prototype distance $d_{t,k}$ generally have different numerical scales. RegimeUnfold therefore normalizes each evidence term before combining them. For a nonnegative evidence term $v_{t,k}$, the median normalization is defined as

$$\text{Norm}(v_{t,k}) = \frac{v_{t,k}}{\text{median}_{t,k}(v_{t,k}) + \epsilon_{\text{norm}}}, \tag{31}$$

where $\epsilon_{\text{norm}} > 0$ is a small constant for numerical stability. The median is computed separately for each type of evidence, placing the predictive, directional, and representation-level terms on comparable scales without changing their within-term ordering.

The normalized evidence is then combined into the local assignment cost

$$\psi_{t,k} = \alpha_e \,\text{Norm}(\bar{e}_{t,k}) + \alpha_b \,\text{Norm}(\xi^2_{t,k}) + \alpha_p \,\text{Norm}(d_{t,k}), \tag{32}$$

where α_e, α_b, and α_p control the relative contributions of predictive error, directional bias, and prototype compatibility, respectively. A smaller $\psi_{t,k}$ indicates stronger local compatibility between sample t and regime k.

A.2.2 HARD REGIME-PATH DECODING

Local assignment evidence alone may still yield fragmented regime paths because each $\psi_{t,k}$ describes compatibility at an individual time point. RegimeUnfold therefore combines the local costs with temporal regularization and decodes the complete hard regime sequence as

$$\widehat{\mathbf{z}} = \arg\min_{\mathbf{z}} \left[\sum_t \psi_{t,z_{t+1}} + \kappa_{\text{sw}} \sum_t \mathbb{I}(z_{t+1} \neq z_t) + \sum_{\mathcal{G} \in \text{Seg}(\mathbf{z})} \phi(|\mathcal{G}|) \right], \tag{33}$$

where $\mathbb{I}(\cdot)$ is the indicator function, $\text{Seg}(\mathbf{z})$ denotes the set of contiguous regime segments in path $\mathbf{z}$, and $|\mathcal{G}|$ is the length of segment $\mathcal{G}$. The switch penalty κ_{sw} discourages unnecessary regime changes,

while the soft duration penalty $\phi(\cdot)$ penalizes excessively short regime segments without imposing a hard minimum duration.

The optimization in Eq. (33) is solved by segment-level dynamic programming. In contrast to independently assigning each sample to its locally best regime, this procedure jointly considers sample-level compatibility, regime switching, and segment persistence when determining the complete hard regime path.

A.2.3 JOINT LEARNING AND ADAPTIVE PATH REFINEMENT

Regime assignments and predictive dynamics are optimized through a hard-EM-style alternating procedure. Given a decoded regime path $\mathbf{z}$, the assignments are first held fixed and the dynamics model is updated by minimizing the composite objective $\mathcal{L}_{\mathrm{dyn}}$ in Eq. (16). The updated dynamics model is then used to recompute the local assignment costs in Eq. (32) and re-decode the complete regime path using Eq. (33). Repeating these two steps progressively aligns the regime-conditioned predictors with the inferred hard assignments.

Once the fixed-K solution satisfies the maturity requirements used for structural search, RegimeUnfold further performs adaptive path refinement. Multiple switch-penalty settings are evaluated to generate alternative hard regime paths under the same regime number. Promising paths are compared according to their predictive performance and path compactness, allowing the model to refine regime boundaries without changing K and reducing sensitivity to a single predefined temporal-smoothness setting.

Offline regime decoding uses the observed next-step targets to evaluate regime-specific predictive compatibility and additionally exploits sequence-level temporal evidence. It is therefore restricted to offline regime discovery and annotation, while online regime inference operates without access to future targets.

A.3 ADAPTIVE REGIME-SPACE CONSTRUCTION

This subsection provides the detailed procedures for generating structural candidates, evaluating proposed regime-space changes, and selecting the final regime number after offline structural search.

A.3.1 STRUCTURAL PROPOSAL GENERATION

Given a sufficiently mature fixed-K solution, RegimeUnfold explores alternative regime-space structures through split, merge, and same-K refinement. These operations respectively expand, contract, or reorganize the current regime space. Each operation first generates a structural proposal and initializes the corresponding candidate model. The candidate is then fully trained and its hard regime path is re-decoded before entering the unified acceptance procedure described in the next subsection.

Split proposals. A split is considered when the observations assigned to a regime exhibit heterogeneous predictive dynamics that may be better represented by two separate regimes. For each eligible parent regime k, RegimeUnfold examines its assigned observations from multiple complementary views, including statistical behavior, temporal dynamics, and latent representations, and generates candidate binary partitions. Using complementary views reduces reliance on a single partitioning criterion and allows different forms of within-regime heterogeneity to be exposed.

Candidate partitions are filtered before model training to remove structurally infeasible or redundant proposals, such as partitions that produce insufficiently supported child groups or fail to provide a meaningful reorganization of the parent regime. For each retained proposal, one child inherits the parent predictor, while the additional regime is initialized from the same parent dynamics and then adapted to the observations assigned to the proposed child partition. Its prototype and bias are initialized from the corresponding child samples. The resulting proposal therefore produces a $(K+1)$-regime candidate that preserves the previously learned parent dynamics while allowing the two child regimes to specialize during subsequent retraining and re-decoding.

Merge proposals. A merge is considered when two existing regimes appear to represent redundant predictive dynamics. RegimeUnfold identifies promising regime pairs primarily through predictive

substitutability. For regimes i and j, the directional cross-regime prediction gaps are defined as

$$\Delta_{i\to j} = \frac{\sum_{t:z_t=i} w_t \,(e_{t,j} - e_{t,i})}{\sum_{t:z_t=i} w_t}, \qquad \Delta_{j\to i} = \frac{\sum_{t:z_t=j} w_t \,(e_{t,i} - e_{t,j})}{\sum_{t:z_t=j} w_t}, \tag{34}$$

where $e_{t,k}$ denotes the prediction error of regime k and w_t downweights observations close to regime transitions. Thus, $\Delta_{i\to j}$ measures the additional predictive loss incurred when observations stably assigned to regime i are instead explained by regime j, and vice versa. Small gaps in both directions indicate that the two regimes can substitute for one another with little predictive degradation.

Predictive substitutability is combined with additional structural similarity to rank merge candidates:

$$S_{ij}^{\text{merge}} = \lambda_c \,\text{Norm}\big(\Delta_{ij}^{\text{cross}}\big) + \lambda_y \,\text{Norm}\Big(D_{ij}^{\text{pred}}\Big) + \lambda_p \,\text{Norm}\big(D_{ij}^{\text{proto}}\big) + \lambda_b \,\text{Norm}\big(D_{ij}^{\text{bias}}\big) - \lambda_a A_{ij}, \tag{35}$$

where $\Delta_{ij}^{\text{cross}}$ summarizes the two directional cross-regime gaps, D_{ij}^{pred}, D_{ij}^{proto}, and D_{ij}^{bias} measure pairwise dissimilarity in predictions, prototypes, and regime biases, respectively, and A_{ij} measures temporal adjacency between the two regimes. The normalization prevents any individual criterion from dominating solely because of its numerical scale. A smaller S_{ij}^{merge} indicates a more plausible merge.

For a selected pair, the merged prototype and bias are initialized by usage-weighted averaging of the two original regimes, while the residual predictor of the more frequently used regime is retained as the starting predictor. The resulting $(K-1)$-regime candidate is then retrained and re-decoded so that the merged regime and the remaining regimes can readjust to the new structure.

Same-K refinement. Same-K refinement is used when the current number of regimes need not change but the partition itself can still be improved. RegimeUnfold supports two forms of such reorganization. First, a provisional split may expose a useful redistribution of observations even when the additional child regime fails to obtain sufficient persistent support. In this case, the unsupported regime is removed while the revised assignments are retained, producing an alternative K-regime candidate. Second, a provisional split may temporarily expand the model to $K+1$ regimes and expose a better partition, after which merging a different pair of regimes returns the model to K regimes with a reorganized structure. These two mechanisms allow the model to revise earlier regime assignments and predictive roles without requiring a permanent change in regime number.

Algorithm 1 summarizes the structural proposal procedure. The three operations generate candidates with different regime-space structures, but no proposal is accepted at this stage. Each candidate must first be fully retrained and re-decoded and is subsequently assessed by the unified evaluation procedure.

A.3.2 Unified candidate evaluation

Let $\mathcal{M}$ denote the mature pre-change model and $\widetilde{\mathcal{M}}$ a candidate obtained through split, merge, or same-K refinement. After $\widetilde{\mathcal{M}}$ is fully trained and re-decoded, the proposed structural change is evaluated according to three criteria: global-score improvement, structural reliability, and voting support over fixed local regime segments.

Global-score criterion. For sequence s, predictive performance is measured from the assigned-regime prediction errors as

$$P_s = \frac{1}{N_s} \sum_{t\in s} \bar{e}_{t,z_{t+1}}, \tag{36}$$

where N_s is the number of valid one-step samples in sequence s and $\bar{e}_{t,z_{t+1}}$ is the predictive error of the assigned regime defined in Section A.2. Regime-path quality is measured by

$$T_s = \lambda_{\text{bd}} \frac{B_s \log N_s}{N_s} + \lambda_{\text{dur}} \frac{\Phi_s}{N_s} + \lambda_{\text{cmp}} \frac{K_s \log N_s}{N_s} + \lambda_{\text{small}} \frac{1}{K_s} \sum_{k=1}^{K_s} \frac{[u_{\min} - u_{s,k}]_+}{u_{\min}}, \tag{37}$$

where B_s is the number of regime boundaries, Φ_s is the accumulated short-segment penalty, K_s is the number of active regimes, and $u_{s,k}$ is the usage of regime k in sequence s. The four terms

Algorithm 1 Structural proposal generation

Require: Mature fixed-K model $\mathcal{M}_K$
Ensure: Candidate set $\mathcal{C}$ for unified structural evaluation

```
Initialize C ← ∅
for each eligible parent regime k do
    Generate binary split proposals from complementary views
    Remove infeasible or redundant partitions
    for each retained split proposal do
        Initialize a (K + 1)-regime candidate from the parent regime
        Add the candidate to C
    end for
end for
for each eligible regime pair (i, j) do
    Compute directional cross-regime prediction gaps
    Evaluate predictive, prototype, bias, and temporal adjacency
    Rank the pair using S_ij^merge
end for
for each retained merge proposal do
    Initialize a (K − 1)-regime candidate from the selected pair
    Add the candidate to C
end for
for each qualifying split-based reorganization do
    if the additional split regime lacks sufficient persistent support then
        Remove the unsupported regime and form a same-K candidate
        Add the candidate to C
    else if merging a different pair yields a reorganized K-regime structure then
        Form a split-then-merge same-K candidate
        Add the candidate to C
    end if
end for
for each candidate M̃ ∈ C do
    Fully train M̃
    Re-decode its complete hard regime path
    Pass M̃ to unified candidate evaluation
end for
```

respectively penalize excessive regime switching, fragmented short segments, unnecessary structural complexity, and insufficient regime usage.

Because P_s and T_s may have different numerical scales, they are rescaled before combination. To distinguish this operation from the median normalization used for local regime scoring, let $\mathrm{Norm}_{\mathrm{str}}(\cdot)$ denote the structural-score normalization. The sequence-level score is $J_s = \lambda_P \mathrm{Norm}_{\mathrm{str}}(P_s) + \lambda_T \mathrm{Norm}_{\mathrm{str}}(T_s)$, where λ_P and λ_T control the relative contributions of predictive performance and regime-path quality. To reduce decision bias caused by incomplete regime coverage in the validation sequence, RegimeUnfold evaluates both the training and validation sequences. For model $\mathcal{M}$, the overall score is

$$J(\mathcal{M}) = \frac{\omega_{\mathrm{tr}} J_{\mathrm{tr}}(\mathcal{M}) + \omega_{\mathrm{val}} J_{\mathrm{val}}(\mathcal{M})}{\omega_{\mathrm{tr}} + \omega_{\mathrm{val}}}, \qquad \omega_{\mathrm{val}} > \omega_{\mathrm{tr}}, \tag{38}$$

where $J_{\mathrm{tr}}(\mathcal{M})$ and $J_{\mathrm{val}}(\mathcal{M})$ are the sequence-level scores computed on the training and validation sequences, respectively. A smaller $J(\mathcal{M})$ indicates better overall performance. The global-score criterion is satisfied when the proposed structural change yields the required operation-specific improvement over the pre-change model.

Structural reliability. Improvement in the global score alone does not guarantee that the resulting regime-space structure is reliable. RegimeUnfold therefore requires every active regime in $\widetilde{\mathcal{M}}$ to have sufficient effective usage and enough valid persistent segments. These requirements prevent

candidates containing weakly supported, highly fragmented, or effectively unused regimes from being accepted solely because they achieve a lower global score. A candidate passes the structural-reliability criterion only when all of its active regimes satisfy the corresponding structural-health requirements.

Voting over fixed local regime segments. The third criterion examines whether the improvement produced by the structural change is consistently supported across the sequence rather than concentrated in a small number of local regions. A fixed set of local regime segments $\mathcal{J}$ is extracted from the mature pre-change regime path, and both $\mathcal{M}$ and $\widetilde{\mathcal{M}}$ are evaluated using the same segment boundaries. This fixed segmentation ensures that the two models are compared on identical local regions.

For each segment $j \in \mathcal{J}$, the relative predictive gain of the candidate is defined as

$$g_j = \frac{\ell_j(\mathcal{M}) - \ell_j(\widetilde{\mathcal{M}})}{|\ell_j(\mathcal{M})|}, \tag{39}$$

where $\ell_j(\cdot)$ denotes the predictive loss evaluated on segment j. A segment supports the proposed structural change when $g_j \geq \delta_{\mathrm{op}}$, where δ_{op} is the operation-specific minimum predictive gain. The weighted support ratio is then

$$\nu_{\mathrm{seg}} = \frac{\sum_{j \in \mathcal{J}} w_j \mathbb{I}[g_j \geq \delta_{\mathrm{op}}]}{\sum_{j \in \mathcal{J}} w_j}, \tag{40}$$

where w_j denotes the weight assigned to segment j. The voting criterion is satisfied when $\nu_{\mathrm{seg}} \geq \tau_{\mathrm{op}}$, where τ_{op} is the operation-specific minimum support ratio.

A proposed split, merge, or same-K refinement is accepted only when the global-score criterion, structural-reliability criterion, and voting criterion are all satisfied. An accepted candidate is retained for subsequent structural search.

A.3.3 Termination and cross-K selection

The structural search terminates when a complete search cycle produces no accepted split, merge, or same-K refinement, or when the regime-bank capacity is reached. The accepted models obtained at different regime numbers are then compared to determine the final offline regime space.

For an accepted model $\mathcal{M}_K$ with K regimes, RegimeUnfold first applies a prediction guard against simpler accepted models. Let

$$P_{\mathrm{val}}^{\mathrm{ref}}(K) = \min_{K' < K} P_{\mathrm{val}}(\mathcal{M}_{K'}) \tag{41}$$

denote the best validation predictive score among accepted models with fewer than K regimes. The candidate $\mathcal{M}_K$ passes the prediction guard when

$$P_{\mathrm{val}}(\mathcal{M}_K) \leq P_{\mathrm{val}}^{\mathrm{ref}}(K) + \tau_{\mathrm{pred}}, \tag{42}$$

where τ_{pred} is the allowed predictive tolerance. This guard prevents a more complex regime-space structure from being retained when its validation predictive performance is noticeably worse than that of a simpler accepted model.

Among the models that pass the prediction guard, let $J_{\min} = \min_{\mathcal{M}} J(\mathcal{M})$ denote the lowest global score. RegimeUnfold then defines the near-best set as

$$\mathcal{C}_{\mathrm{near}} = \{\mathcal{M} : J(\mathcal{M}) \leq J_{\min} + \tau_J\}, \tag{43}$$

where τ_J is determined by the prescribed absolute and relative tolerances. The model with the smallest regime number within $\mathcal{C}_{\mathrm{near}}$ is selected as the final offline model. This rule avoids unnecessary regime-space expansion when a larger model provides only marginal improvement in the global score.

B Details of Online Regime Evolution

This section provides the detailed formulations and procedures underlying the three main parts of online regime evolution: regime selection and forecasting within the current regime space, evidence-driven regime-space evolution assessment, and stability-aware regime-space adaptation.

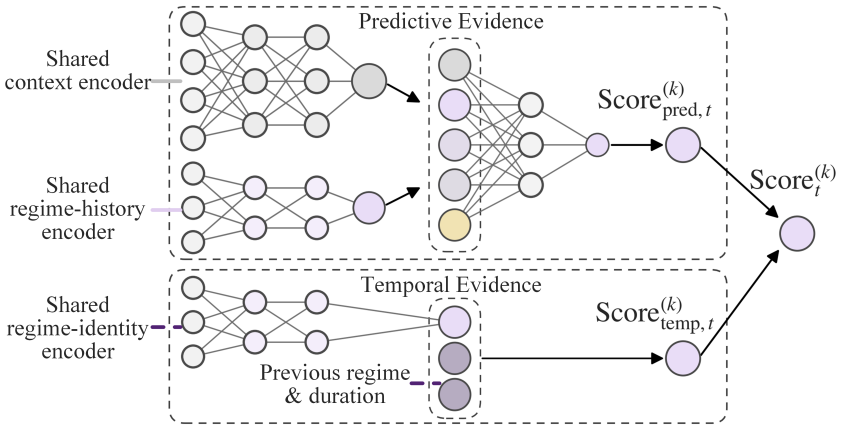


Figure 3: Structure of the online regime selection router. Predictive and temporal evidence are encoded separately, fused adaptively, and used for hard regime decoding and regime-conditioned forecasting.

B.1 Regime Selection and Forecasting within the Current Regime Space

This subsection provides the detailed construction, training, and online operation of the regime selection router introduced in Section 4.2. The router selects the next regime from the current regime space $\mathcal{R}_t$ using only information available up to time t. Its predictive and temporal branches use shared parameters across regimes, so the routing architecture does not depend on the current number of regimes. An overview of the router is shown in Figure 3, which highlights how predictive and temporal evidence are encoded separately, fused adaptively, and used for hard regime decoding and regime-conditioned forecasting.

B.1.1 Predictive and temporal routing evidence

For predictive routing, RegimeUnfold maintains a recent history window $\mathcal{W}_t$ ending at time t. At each observed time $\tau \in \mathcal{W}_t$, the global predictive context contains the recent observation, its latent representation, and the prediction residual of the shared-base predictor. Let $\mathbf{a}_\tau$ denote the resulting global-context feature vector. A shared context encoder aggregates these features across the history window into

$$\mathbf{o}_t = E_{\text{ctx}}\left(\{\mathbf{a}_\tau\}_{\tau\in\mathcal{W}_t}\right). \tag{44}$$

For each regime $k \in \mathcal{R}_t$, the router additionally maintains regime-specific predictive evidence over the same history window. Let

$$\boldsymbol{\delta}_{\tau,k} = \mathbf{x}_\tau - \widehat{\mathbf{x}}^{(k)}_\tau \tag{45}$$

denote the realized prediction residual of regime k. The regime-specific evidence summarizes signed output-wise deviations, squared prediction errors, mean and worst-output errors, relative predictive regret with respect to competing regimes, and prototype compatibility. Let $\mathbf{r}_{\tau,k}$ denote the resulting regime-specific evidence vector. A shared regime-history encoder combines its recent sequence with the corresponding regime prototype $\mathbf{p}_k$:

$$\mathbf{u}_{t,k} = E_{\text{hist}}\left(\{\mathbf{r}_{\tau,k}\}_{\tau\in\mathcal{W}_t}, \mathbf{p}_k\right). \tag{46}$$

The same encoder is applied to every regime, and the predictive branch does not directly use the previously inferred regime or its duration. A shared scoring function evaluates each regime relative to the current predictive context, and the resulting scores are normalized across the available regimes:

$$\text{Score}^{(k)}_{\text{pred},t} = \frac{\exp(S_{\text{pred}}\left(\mathbf{o}_t, \mathbf{u}_{t,k}\right))}{\sum_{j\in\mathcal{R}_t}\exp(S_{\text{pred}}\left(\mathbf{o}_t, \mathbf{u}_{t,j}\right))}. \tag{47}$$

Thus, the predictive branch evaluates all regimes relative to the same current context while retaining their regime-specific recent predictive behavior.

The temporal branch is constructed separately to represent persistence information without directly mixing it into the predictive representation. A shared regime-identity encoder maps each regime prototype to

$$\mathbf{e}_k^{\mathrm{id}} = E_{\mathrm{id}}(\mathbf{p}_k). \tag{48}$$

When a reliable previous-regime context is available, let z_t^{ctx} denote the previously inferred regime and let ℓ_t denote its current duration. For each regime $k \in \mathcal{R}_t$, the temporal input is constructed from the previous-regime identity, the identity of regime k, their difference, a same-regime indicator, and the current duration:

$$\mathbf{q}_{t,k}^{\mathrm{temp}} = \left[\mathbf{e}_{z_t^{\mathrm{ctx}}}^{\mathrm{id}}, \mathbf{e}_k^{\mathrm{id}}, \mathbf{e}_k^{\mathrm{id}} - \mathbf{e}_{z_t^{\mathrm{ctx}}}^{\mathrm{id}}, \mathbb{I}(k = z_t^{\mathrm{ctx}}), \log(1+\ell_t)\right]. \tag{49}$$

A shared temporal scoring function evaluates the consistency of each regime with the available temporal context, and the resulting scores are normalized across the current regime space:

$$\mathrm{Score}_{\mathrm{temp},t}^{(k)} = \frac{\exp\left(S_{\mathrm{temp}}\left(\mathbf{q}_{t,k}^{\mathrm{temp}}\right)\right)}{\sum_{j\in\mathcal{R}_t}\exp\left(S_{\mathrm{temp}}\left(\mathbf{q}_{t,j}^{\mathrm{temp}}\right)\right)}. \tag{50}$$

When the previous-regime context is not considered reliable, its identity- and duration-dependent inputs are masked rather than forcing an uncertain regime into the temporal branch. The temporal scores therefore provide complementary evidence about the consistency of each regime with the currently available temporal context.

B.1.2 Adaptive fusion and router training

The normalized predictive and temporal scores are fused adaptively. Let $\mathbf{q}_t^{\mathrm{pred}}$ denote the current routing context derived from the predictive branch, $\mathbf{g}_t^{\mathrm{set}}$ an aggregate representation of the regimes in $\mathcal{R}_t$, $\mathbf{e}_t^{\mathrm{prev}}$ the available previous-regime identity context, and $\boldsymbol{\chi}_t$ confidence and agreement statistics derived from the predictive and temporal scores. The fusion gate produces

$$\alpha_t = \sigma\left(F_\alpha\left(\mathbf{q}_t^{\mathrm{pred}}, \mathbf{g}_t^{\mathrm{set}}, \mathbf{e}_t^{\mathrm{prev}}, \boldsymbol{\chi}_t\right)\right), \qquad \alpha_t \in [0,1]. \tag{51}$$

For each regime $k \in \mathcal{R}_t$, the final routing score is

$$\mathrm{Score}_t^{(k)} = (1-\alpha_t)\mathrm{Score}_{\mathrm{pred},t}^{(k)} + \alpha_t \mathrm{Score}_{\mathrm{temp},t}^{(k)}. \tag{52}$$

A smaller α_t places greater emphasis on predictive evidence, whereas a larger value increases the contribution of temporal evidence. The confidence and agreement inputs allow the fusion gate to adjust this balance according to the reliability and consistency of the two branches at the current time.

The regime selection router is trained before online deployment using the regime annotations obtained from offline discovery. Let z_{t+1}^{off} denote the offline regime annotation and $C_{t,k}$ the one-step prediction cost associated with regime k for sample t. The routing objective is

$$\mathcal{L}_{\mathrm{router}} = -\lambda_{\mathrm{route}} \sum_t \log \mathrm{Score}_t^{(z_{t+1}^{\mathrm{off}})} + \lambda_{\mathrm{cost}} \sum_t \sum_{k\in\mathcal{R}_t} \mathrm{Score}_t^{(k)} C_{t,k}. \tag{53}$$

The first term encourages consistency with the regime structure learned during offline discovery, while the second provides prediction-aware guidance by assigning lower cost to routing decisions associated with more accurate regime-conditioned forecasts. The predictive branch, temporal branch, and fusion gate are jointly optimized through the final routing objective rather than being trained as independent classifiers. The historical sequence available before online deployment is divided chronologically into training and development portions for router learning, with the latter used to select the routing model while preserving causal ordering.

B.1.3 Hard decoding and online context update

At forecasting time, the router uses only information available through time t and performs hard regime selection:

$$\widehat{z}_{t+1} = \arg\max_{k\in\mathcal{R}_t} \mathrm{Score}_t^{(k)}, \qquad \widehat{\mathbf{x}}_{t+1} = \widehat{\mathbf{x}}_{t+1}^{(\widehat{z}_{t+1})}. \tag{54}$$

Only the predictor associated with $\widehat{z}_{t+1}$ is used for forecasting, so regime-conditioned predictions are not averaged across multiple regimes.

After $\mathbf{x}_{t+1}$ becomes available, the newly realized regime-specific predictive evidence is appended to the histories used by the router. The observed regime context and its duration are then updated causally using the observation prefix available through time $t+1$ together with temporal continuity. The resulting z_{t+1}^{ctx} and ℓ_{t+1} are used only when forecasting the subsequent observation $\mathbf{x}_{t+2}$. Therefore, $\mathbf{x}_{t+1}$ is never used to select $\widehat{z}_{t+1}$ itself and enters the routing process only after it has been observed.

B.2 Evidence-Driven Regime-Space Evolution Assessment

This subsection provides the detailed construction, supervision, and online assessment procedure of the regime evolution router introduced in Section 4.2. After each new observation becomes available, the router compares the observed dynamics with regime-specific predictive and structural references and produces two complementary evidence scores for assessing whether the current regime space remains adequate.

B.2.1 Regime-specific predictive and structural references

For each established regime $k \in \mathcal{R}_t$, RegimeUnfold maintains predictive and structural references derived from observations associated with that regime. The predictive reference summarizes the historical prediction-error level of the corresponding regime-conditioned predictor. Let

$$e_{t,k} = \frac{1}{d} \left\| \mathbf{x}_t - \widehat{\mathbf{x}}_t^{(k)} \right\|_2^2 \tag{55}$$

denote the realized one-step prediction error of regime k after $\mathbf{x}_t$ becomes available, and let e_k^{ref} denote its historical predictive reference computed from established observations associated with regime k. The relative predictive discrepancy is represented as

$$r_{t,k}^{\mathrm{pred}} = \log\left(1 + \frac{e_{t,k}}{e_k^{\mathrm{ref}} + \epsilon}\right), \tag{56}$$

where $\epsilon > 0$ is a small constant for numerical stability. This relative formulation evaluates current predictive degradation with respect to the normal error scale of each regime rather than directly comparing raw errors across regimes.

Structural references provide complementary information about whether the newly observed dynamics remain similar to those previously associated with each regime. Let $\boldsymbol{\phi}_t$ denote a structural descriptor extracted from the recent observation context. The descriptor characterizes the observed dynamics from multiple views, including level, scale, trend, temporal dynamics, distributional behavior, and spectral characteristics. For each regime k, the corresponding structural reference $\mathcal{S}_k^{\mathrm{ref}}$ summarizes these characteristics over its established observations. The structural discrepancy is written compactly as

$$\mathbf{r}_{t,k}^{\mathrm{str}} = D_{\mathrm{str}}\left(\boldsymbol{\phi}_t, \mathcal{S}_k^{\mathrm{ref}}\right), \tag{57}$$

where $D_{\mathrm{str}}(\cdot)$ denotes the normalized multi-view deviation from the regime-specific structural reference. The predictive and structural references therefore characterize two complementary forms of mismatch: degradation of regime-conditioned forecasting and departure from previously established regime structure.

B.2.2 Shared evidence representation and dual-score learning

For each regime $k \in \mathcal{R}_t$, the predictive and structural discrepancies are combined into

$$\mathbf{d}_{t,k} = \left[r_{t,k}^{\mathrm{pred}}, \mathbf{r}_{t,k}^{\mathrm{str}}\right]. \tag{58}$$

The regime-wise evidence is then summarized into a fixed-dimensional set-level representation,

$$\mathbf{v}_t = A_{\mathrm{evo}}\left(\{\mathbf{d}_{t,k}\}_{k \in \mathcal{R}_t}\right), \tag{59}$$

which provides a common representation of how the current observation relates to the entire available regime space. From $\mathbf{v}_t$, the evolution router produces two complementary evidence scores,

$$\begin{bmatrix} \mathrm{Score}_{\mathrm{ins},t} \\ \mathrm{Score}_{\mathrm{nov},t} \end{bmatrix} = \sigma\left(G_\eta(\mathbf{v}_t)\right), \tag{60}$$

where the sigmoid is applied elementwise. The two outputs are independently learned task scores rather than a softmax distribution over mutually exclusive outcomes. Predictive insufficiency and structural novelty may therefore occur simultaneously.

The two scores share the same evidence representation but use different supervision. Let

$$\rho_t^{\mathrm{best}} = \min_{k \in \mathcal{R}_t} \frac{e_{t,k}}{e_k^{\mathrm{ref}} + \epsilon} \tag{61}$$

denote the predictive degradation of the best available regime relative to its historical reference. Predictive-insufficiency labels are constructed from this relative degradation so that positive samples correspond to observations for which none of the available regime predictors provides a sufficiently reliable explanation.

Structural-novelty supervision is constructed through hide-one-regime simulation because genuinely unseen regimes are unavailable during offline training. For each hide-one training episode, one annotated regime $k^\star$ is removed from the available regime set, and all predictive and structural information associated with $k^\star$ is excluded when constructing the evidence representation. Eligible observations governed by $k^\star$ provide positive novelty supervision, while observations governed by the remaining visible regimes provide negative supervision. Ambiguous hidden-regime observations that remain sufficiently well explained by the visible regimes are excluded from positive novelty supervision. This construction encourages structural novelty to represent departure from the available regime space rather than merely large prediction error.

The two tasks are jointly optimized through the weighted binary cross-entropy objective

$$\mathcal{L}_{\mathrm{evo}} = -\sum_t \sum_{q \in \{\mathrm{ins},\mathrm{nov}\}} w_{t,q} \left[y_{t,q} \log \mathrm{Score}_{q,t} + (1 - y_{t,q}) \log(1 - \mathrm{Score}_{q,t}) \right], \tag{62}$$

where $y_{t,q}$ denotes the task-specific supervision and $w_{t,q}$ balances the corresponding training samples. Sharing the evidence representation allows the two tasks to exploit common information about changing dynamics while retaining distinct learning objectives.

B.2.3 Online evidence activation and persistence

During online operation, the two evidence scores are assessed using independently calibrated thresholds,

$$\mathrm{Score}_{\mathrm{ins},t} \geq \tau_{\mathrm{ins}}, \qquad \mathrm{Score}_{\mathrm{nov},t} \geq \tau_{\mathrm{nov}}, \tag{63}$$

where the thresholds are calibrated separately because predictive insufficiency and structural novelty represent different forms of regime-space mismatch rather than complementary probabilities.

A high predictive-insufficiency score contributes to regime-space evolution assessment only when it is also supported by sufficiently large realized prediction degradation. Structural-novelty evidence is evaluated over consecutive observations so that an isolated structural deviation does not by itself indicate persistent evolution. Evidence that is not sustained is allowed to decay, whereas persistent predictive insufficiency or structural novelty provides increasing support that the current regime space no longer adequately explains the incoming dynamics.

Only sufficiently supported evidence is passed to the stability-aware regime-space adaptation procedure described in the following subsection. The regime evolution router therefore assesses the need for adaptation but does not itself determine whether an existing regime should be updated or a new regime should be created.

B.3 Stability-Aware Regime-Space Adaptation

This subsection provides the detailed procedures for constructing, evaluating, and integrating the UPDATE and CREATE alternatives introduced in Section 4.2. Once sufficiently persistent evolution evidence has accumulated, RegimeUnfold forms an adaptation episode from the corresponding

streaming observations. Rather than immediately modifying the deployed model, the episode is used to examine whether the newly observed dynamics can be absorbed by an existing regime or require expansion of the regime space.

B.3.1 Adaptation episode and candidate construction

Let $\mathcal{E}_t$ denote the adaptation episode accumulated up to time t. The observations in $\mathcal{E}_t$ provide the common evidence used to construct both UPDATE and CREATE alternatives. Candidate training uses only observations that have already arrived, while part of the available evidence is withheld from parameter fitting for model selection and stability assessment. In addition, representative observations from established regimes are retained as historical references so that adaptation to the recent episode can be evaluated without losing previously learned predictive behavior.

For UPDATE, RegimeUnfold first identifies a small set of existing regimes that remain plausible explanations of the adaptation episode. These regimes are examined individually rather than being modified jointly. For a candidate regime k, its regime-specific predictive component is adapted to the newly observed dynamics while the shared predictive structure and the remaining established regimes are preserved. The resulting candidate is denoted by $\mathcal{M}_t^{\mathrm{upd},k}$. After the plausible regimes have been evaluated, the strongest validated regime-specific adaptation is retained as the UPDATE alternative $\mathcal{M}_t^{\mathrm{upd}}$. This procedure tests whether the observed change can be represented as drift within an existing regime without altering the size of the regime space.

For CREATE, RegimeUnfold introduces an additional regime-specific predictor and trains it using the same adaptation episode. The new predictor is initialized from compatible established dynamics and is allowed to specialize toward the accumulated observations, while the existing regime predictors are protected from unnecessary modification. The resulting candidate $\mathcal{M}_t^{\mathrm{new}}$ represents the alternative hypothesis that the incoming dynamics cannot be adequately explained by adapting any existing regime and therefore require an additional regime.

Because the adaptation decision may remain unresolved while the current regime predictors already produce large forecasting errors, RegimeUnfold additionally maintains a lightweight provisional predictor using the recent adaptation episode. The provisional predictor temporarily supports forecasting during this unresolved period, but is not treated as an established regime and does not directly alter the regime space.

B.3.2 UPDATE-CREATE competition and certification

Once trained, the UPDATE and CREATE alternatives are evaluated under a common protocol from two aspects: predictive performance and stability preservation. Predictive performance assesses whether the candidate improves forecasting on the accumulated adaptation episode and whether this improvement remains supported by held-out evidence that is not used for candidate fitting. This separation prevents an adaptation from being accepted solely because it fits the observations from which it was constructed.

Stability preservation examines whether the improvement on the recent dynamics is obtained without materially degrading established regimes. RegimeUnfold therefore evaluates the candidate on representative historical evidence and checks both regime-specific forecasting performance and the consistency of regime assignments. An adaptation that improves the recent episode but substantially worsens predictions for established regimes or causes excessive reassignment of previously stable observations is rejected.

Among the alternatives satisfying both predictive-performance and stability requirements, the UPDATE and CREATE candidates are compared with the current model. The model with the strongest validated predictive performance is retained. If neither adaptation provides sufficiently supported improvement, the current model is preserved and additional streaming evidence is accumulated.

CREATE is subject to an additional certification requirement because it increases the complexity of the regime space. Beyond predictive improvement and stability preservation, the candidate regime must receive sufficient persistent support from the accumulated observations and remain sufficiently distinct from the existing regimes. These requirements prevent transient deviations or ordinary drift within an established regime from being incorrectly converted into a permanent new regime.

B.3.3 STABLE INTEGRATION AND POST-ADAPTATION UPDATE

An accepted adaptation is incorporated only after the complete evaluation procedure has been satisfied. If UPDATE is selected, the validated regime-specific predictor replaces its previous version while the regime space itself remains unchanged,

$$\mathcal{R}_{t+1} = \mathcal{R}_t. \tag{64}$$

The remaining regime-specific predictors are preserved, and subsequent regime selection and evolution assessment use the updated predictive dynamics of the adapted regime.

If CREATE is selected, the validated predictor and its corresponding regime-specific references are incorporated as a new regime k_{new}, yielding

$$\mathcal{R}_{t+1} = \mathcal{R}_t \cup \{k_{\text{new}}\}. \tag{65}$$

Because the regime selection router uses shared scoring functions across regimes, the new regime can subsequently participate in online selection through its own predictive history and regime references without changing the form of the routing architecture.

After the adaptation decision is resolved, the provisional predictor is removed. If UPDATE or CREATE is accepted, subsequent forecasting proceeds with the adapted model and the corresponding regime space. If neither alternative is accepted, the current model is retained and further streaming evidence is accumulated until a later adaptation attempt is sufficiently supported.

C SYNTHETIC DATASET DESCRIPTIONS

We construct five synthetic datasets from a higher-order semi-Markov regime process with state-dependent autoregressive emissions. The generator contains a single latent regime process shared by multiple output channels, consistent with the forecasting setting considered in RegimeUnfold, where regime changes correspond to changes in the predictive dynamics of a multivariate observation process. Four datasets use a fixed latent regime space with different sequence lengths, output dimensions, and regime numbers, while the fifth progressively expands the available regime set to evaluate regime-space evolution as previously unseen predictive dynamics emerge. One fixed-K dataset additionally uses shortened sojourn times to create a fast-switching scenario.

C.1 DATA-GENERATING PROCESS

Let $\mathbf{x}_t = (x_{t,1}, \ldots, x_{t,d})^\top \in \mathbb{R}^d$ denote the multivariate observation at time t. The time-point regime sequence $\{z_t\}$ is generated from a sequence of regime segments. Let c_n denote the regime of the n-th segment and τ_n its duration. For fixed-regime datasets, $\mathcal{K}_t = \{1, \ldots, K\}$ remains unchanged throughout the sequence. For the Progressive dataset, the active regime set expands at predefined stage boundaries. At the beginning of the initial stage, the active regimes are explicitly introduced in regime-index order before ordinary transition sampling resumes. When the regime space expands, each newly available regime is introduced immediately at the corresponding stage boundary and its sojourn time is sampled from the corresponding regime-specific duration distribution. The higher-order regime history is then reinitialized from the newly introduced regime before subsequent transition sampling resumes. This construction guarantees that every intended regime is represented and that progressive regime-space expansion is actually encountered in the generated sequence.

Higher-order regime transitions. The segment-level latent regime process follows an L-th order transition mechanism,

$$\Pr(c_n = k \mid c_{n-1}, \ldots, c_{n-L}) = \pi_n(k \mid c_{n-1:n-L}, \mathcal{K}_n)\,, \tag{66}$$

where L denotes the transition order and $\mathcal{K}_n$ denotes the active regime set at the starting time of segment n. In the experiments, we use a second-order process with $L = 2$. If the two most recent segment regimes are identical, the next segment remains in the same regime with probability 0.8, while the remaining probability 0.2 is distributed uniformly over the other active regimes. If the two most recent segment regimes differ, the next regime is sampled uniformly from the active regime set. This construction introduces higher-order persistence without requiring a separate transition matrix for each synthetic dataset.

Table 4: Configurations of the five synthetic datasets.

Dataset	T	d	Regime setting	Duration setting
Fixed-$K2$	10,000	2	$K = 2$	Standard
Fast-$K2$	5,000	2	$K = 2$	Fast
Fixed-$K3$	10,000	3	$K = 3$	Standard
Fixed-$K5$	10,000	5	$K = 5$	Standard
Progressive	10,000	3	$2 \to 3 \to 4$ $(0.5/0.2/0.3)$	Standard

Semi-Markov durations. After the regime of segment n is selected as $c_n = k$, its duration is sampled from the regime-specific sojourn-time distribution

$$\tau_n \sim D_k, \qquad \tau_n \geq 1. \tag{67}$$

The time-point regime sequence remains fixed at $z_t = k$ throughout the corresponding segment. Under ordinary evolution, once the sampled duration expires, the next segment regime is generated according to the higher-order transition rule above and a new duration is sampled. In the Progressive dataset, a predefined stage boundary may terminate the current segment early when a newly available regime is explicitly introduced. If the transition rule selects the same regime again, a new duration is sampled and the regime continues without an observable state change. The simulator uses geometric, shifted-Poisson, and fixed-duration distributions. Different duration configurations are used for the standard- and fast-switching settings.

State-dependent autoregressive emissions. Conditional on the latent regime, each output channel follows regime-specific autoregressive dynamics. To match the implementation, we distinguish the latent autoregressive base process from the optional regime bias. For output q,

$$y_{t,q} = \sum_{p=1}^{P} a_{q,z_t}^{(p)} y_{t-p,q} + \varepsilon_{t,q}, \qquad \varepsilon_{t,q} \sim \mathcal{N}(0, \sigma_{z_t}^2), \tag{68}$$

and the observed value is

$$x_{t,q} = y_{t,q} + b_{z_t}, \tag{69}$$

where $a_{q,z_t}^{(p)}$ is the autoregressive coefficient of output q under regime z_t, b_{z_t} is the regime-specific bias, and σ_{z_t} controls the innovation noise. The autoregressive history is maintained on the bias-free base process $y_{t,q}$, so that the regime bias acts as an additive observation shift rather than being recursively propagated through the autoregressive dynamics. The autoregressive history is initialized as empty, and unavailable lagged terms at the beginning of the sequence therefore contribute zero. With $P = 1$, the first base observation consists only of the innovation term. Innovations are sampled independently across time points and output channels. In all five datasets, the emission model uses first-order autoregression, i.e., $P = 1$.

All output channels share the same latent regime sequence but use different regime-specific autoregressive coefficients. Consequently, a regime change modifies the predictive dynamics of multiple outputs simultaneously rather than affecting only a single observation channel.

C.2 Simulation Configurations

We construct five synthetic datasets to evaluate RegimeUnfold under different regime-space complexities, transition frequencies, and regime-space evolution settings. Four datasets have a fixed regime space, while the fifth progressively introduces previously unseen regimes. Table 4 summarizes the configurations used in the experiments.

The Fixed-$K2$ dataset provides the basic fixed-regime setting. Fast-$K2$ uses the same two-regime predictive structure but substantially shorter sojourn times to evaluate performance under frequent regime transitions. Fixed-$K3$ and Fixed-$K5$ increase both the number of latent regimes and the output dimension, thereby increasing the complexity of regime discovery. The Progressive dataset starts with two available regimes and subsequently expands the regime space to three and then four regimes. The three stages occupy 50%, 20%, and 30% of the sequence, corresponding to lengths of

Table 5: Regime-dependent first-order autoregressive coefficients $a_{q,k}^{(1)}$.

Output	Regime 1	Regime 2	Regime 3	Regime 4	Regime 5
1	1.0	−0.9	0.5	−0.2	0.0
2	0.5	1.0	−0.9	1.0	−0.2
3	−0.9	1.0	0.0	−0.8	−0.2
4	1.0	−0.9	−0.9	1.0	−0.2
5	−0.9	1.0	0.5	−0.9	1.0

Table 6: Regime-specific sojourn-time distributions used in the synthetic datasets.

Regime	Standard setting	Fast setting
1	$\text{Geometric}(p = 0.04)$	$\text{Geometric}(p = 0.15)$
2	$1 + \text{Poisson}(30)$	$1 + \text{Poisson}(5)$
3	Fixed at 50	-
4	$1 + \text{Poisson}(70)$	-
5	$1 + \text{Poisson}(30)$	-

5000, 2000, and 3000, respectively. This provides a controlled setting for evaluating online regime-space evolution as previously unseen predictive dynamics emerge.

Figures 4-8 illustrate the five synthetic datasets, with background colors indicating the ground-truth latent regimes.

Autoregressive dynamics. All datasets use first-order autoregressive emissions with regime- and output-specific coefficients. The coefficients used across the five datasets are reported in Table 5. A dataset with d outputs uses the first d rows, and a fixed-K dataset uses the first K regime columns. The Progressive dataset uses the first three outputs and the first four regimes.

The innovation standard deviation is fixed at $\sigma_k = 0.01$ for all regimes. The four fixed-regime datasets use zero regime bias, $b_k = 0$. For the Progressive dataset, the regime biases for regimes 1-4 are $(b_1, b_2, b_3, b_4) = (0.1, -0.1, 0, 0.05)$.

Sojourn-time distributions. The standard and fast-switching duration settings used in the experiments are shown in Table 6. The fast setting is used only for Fast-$K2$ and therefore requires duration distributions only for the first two regimes.

D Real-World Dataset Descriptions

We use four preprocessed real-world datasets: the Server Machine Dataset (SMD), an exchange-rate dataset (Exchange), a traffic-speed dataset (Traffic), and a gold futures price dataset (GOLD). The first three datasets follow the same preprocessing and experimental configurations as DRL-STAF (Jiang et al., 2026), while GOLD is additionally included to evaluate forecasting on commodity-market data. The original datasets remain publicly accessible from their respective sources. Together, these datasets cover server monitoring, foreign exchange, transportation systems, and commodity markets, providing heterogeneous real-world settings for evaluating forecasting and, where reference labels are available, regime inference.

Server Machine Dataset. The Server Machine Dataset (SMD) (Su et al., 2019) is publicly available from the OmniAnomaly repository at `https://github.com/NetManAIOps/OmniAnomaly`. It contains multivariate monitoring time series collected from 28 server machines, with 38 measurements for each machine. The original dataset consists of an unlabeled training portion and a test portion with point-level anomaly labels. Following the preprocessing used in DRL-STAF (Jiang et al., 2026), we use a three-dimensional subset extracted from the labeled test sequence of one server machine, with a sequence length of 7000. The selected sequence contains both normal and anomalous periods. The anomaly labels are retained only as external reference la-

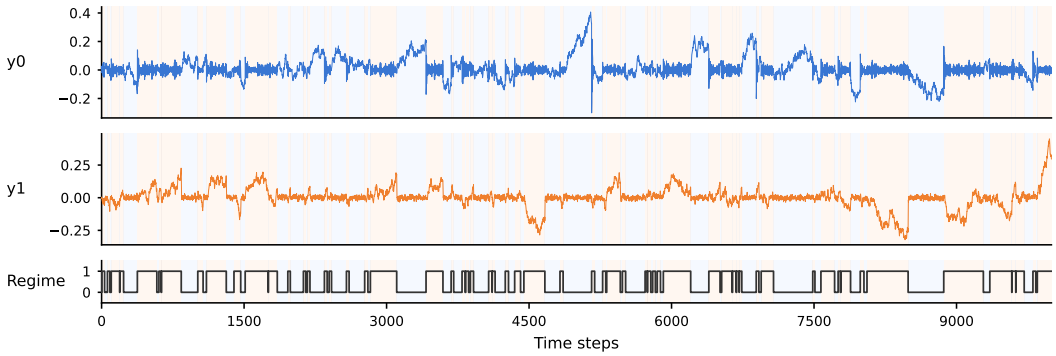


Figure 4: Synthetic dataset (Fixed-$K2$).

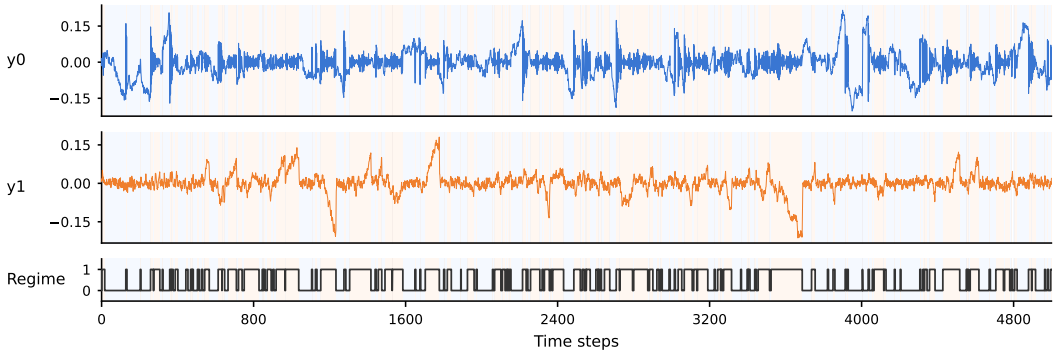


Figure 5: Synthetic dataset (Fast-$K2$).

bels for evaluating regime inference and are not used for model training or regime discovery. Figure 9 shows the selected multivariate sequence and its reference states.

Exchange rate dataset. The Exchange rate dataset follows the preprocessing and experimental configuration used in DRL-STAF (Jiang et al., 2026). The original daily foreign-exchange records are obtained from `https://in.investing.com`. The dataset contains USD/CNY, USD/EUR, and USD/JPY observations from January 2006 to February 2025, with a total sequence length of 5000. The closing-price series of the three exchange rates are jointly modeled as a three-dimensional multivariate time series and are used for next-step forecasting. Since no explicit regime annotations are available, the Exchange dataset is used to evaluate forecasting performance, while regime inference is not quantitatively evaluated. Figure 10 shows the three exchange-rate series used in our experiments.

Traffic speed dataset. The Traffic speed dataset follows the preprocessing used in DRL-STAF (Jiang et al., 2026) and is derived from the traffic data released with StemGNN (Cao et al., 2020), available at `https://github.com/microsoft/StemGNN`. The original data contain multivariate traffic speed measurements collected from highway sensor stations in California. The preprocessed subset used in our experiments contains five neighboring sensor nodes with strong pairwise correlations and a sequence length of 5000. The five sensor series are jointly modeled as the multivariate observation process for next-step forecasting. Since explicit regime annotations are unavailable, the Traffic dataset is used to evaluate forecasting performance, while regime inference is not quantitatively evaluated. Figure 11 shows the five traffic-speed series used in our experiments.

Gold futures dataset. The GOLD dataset consists of daily gold futures price records obtained from `https://in.investing.com`, covering the period from January 1, 2000 to December 17, 2025. Each observation contains four price dimensions: opening price, closing price, highest price, and lowest price. The four series are jointly modeled as a four-dimensional multivariate time

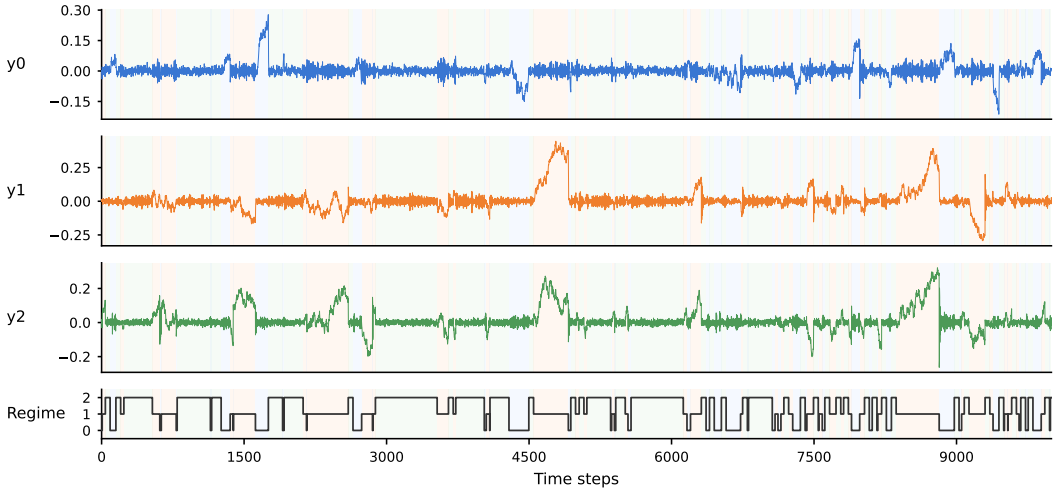


Figure 6: Synthetic dataset (Fixed-$K3$).

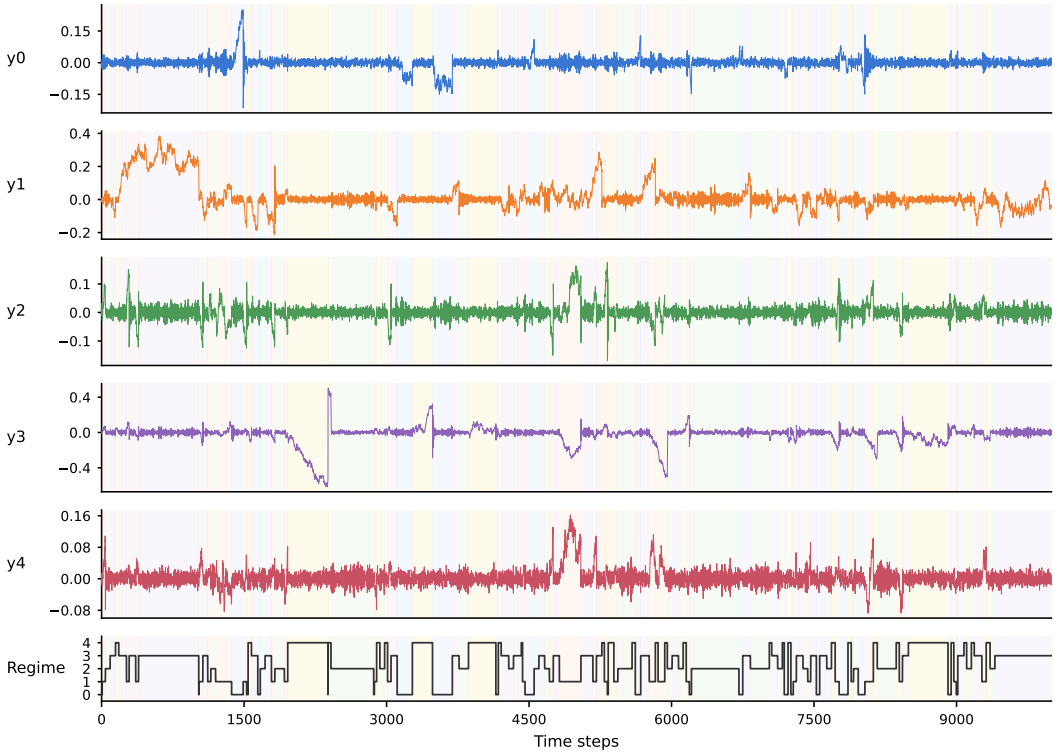


Figure 7: Synthetic dataset (Fixed-$K5$).

series and are used for next-step forecasting. Since explicit regime annotations are unavailable, the GOLD dataset is used to evaluate forecasting performance, while regime inference is not quantitatively evaluated. Figure 12 shows the four gold futures price series used in our experiments.

E Full Results

In this section, we provide supplementary results to further demonstrate the effectiveness of Regime-Unfold across different settings and datasets. For iTransformer, PatchTST, and TimeMixer, we additionally evaluate long-window variants, allowing their native temporal modeling mechanisms to

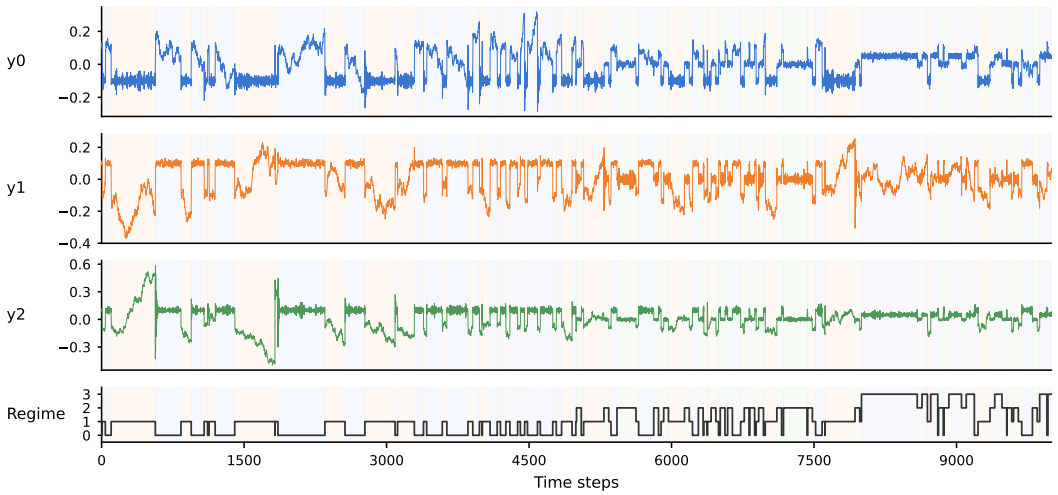


Figure 8: Synthetic dataset (Progressive).

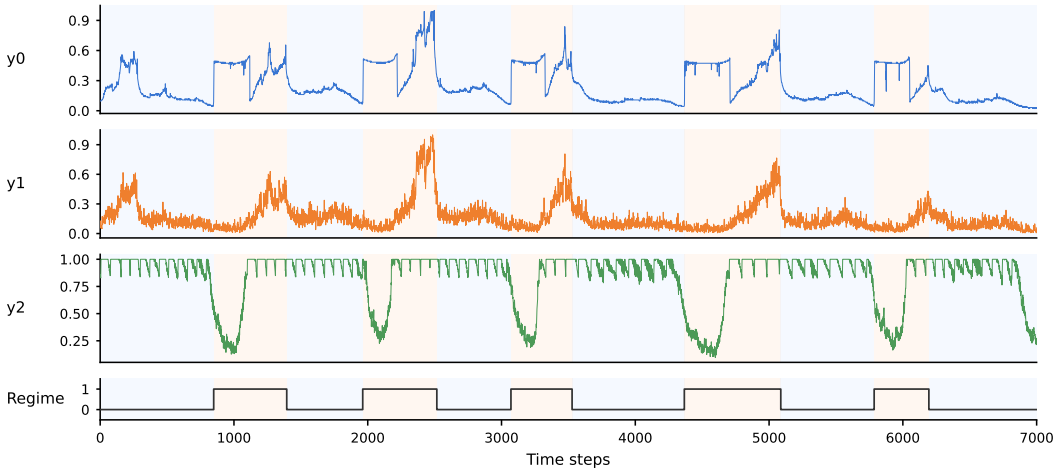


Figure 9: Selected sequence from the Server Machine Dataset.

operate with broader historical contexts. Since RegimeUnfold supports online adaptation whereas many comparison methods do not provide an explicit adaptive setting, we further construct adaptive variants of applicable baselines by periodically fine-tuning them on a sliding window every 16 time steps. This provides a more direct comparison between non-adaptive and adaptive forecasting under the same online evaluation setting.

E.1 Results on Synthetic Datasets

Results on Fixed-$K2$ dataset. Table 7 reports the complete results on the Fixed-$K2$ synthetic dataset, including mean values and standard deviations across repeated runs. RegimeUnfold achieves the best overall performance in both regime inference and forecasting, outperforming both non-adaptive and adaptive baselines as well as their long-window variants. Figure 13 further presents the forecasting and regime inference results. RegimeUnfold closely tracks the observations and accurately recovers most regime transitions. Near the end of the sequence, a temporary change is marked as "Unresolved", indicating that the accumulated evidence is insufficient to confirm a regime update or expansion. Rather than immediately modifying the regime space, RegimeUnfold retains the unresolved status while continuing forecasting, reflecting its conservative evolution mechanism under uncertain changes.

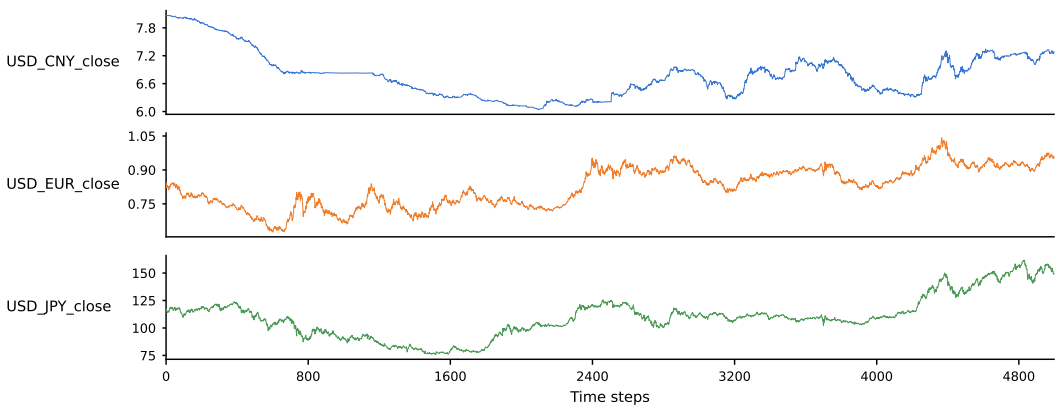


Figure 10: Selected exchange-rate time series.

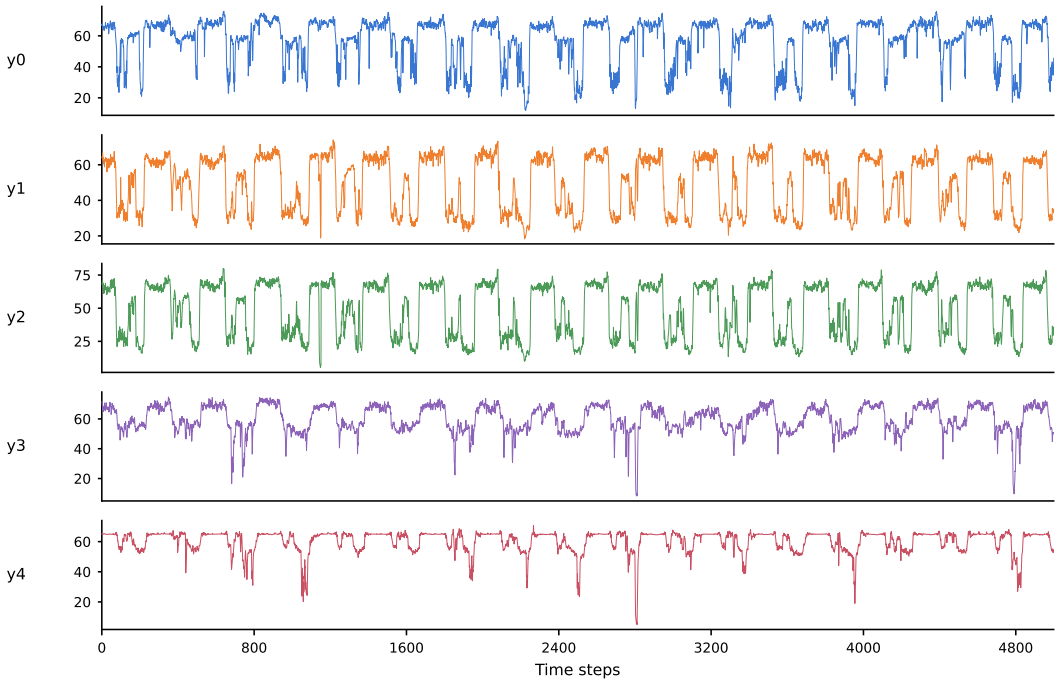


Figure 11: Selected traffic-speed time series.

Results on Fast-$K2$ dataset. Table 8 reports the complete results on the Fast-$K2$ synthetic dataset. RegimeUnfold again achieves the best overall performance in both regime inference and forecasting. Compared with Fixed-$K2$, Fast-$K2$ contains the same two regimes but switches between them more frequently. Accordingly, RegimeUnfold shows moderately lower regime inference accuracy and higher forecasting errors than on Fixed-$K2$, reflecting the increased difficulty caused by more frequent regime transitions. Nevertheless, it maintains clear advantages over the comparison methods, demonstrating robust regime tracking and forecasting under rapidly switching dynamics. Figure 14 further shows that RegimeUnfold closely follows the observations and captures the frequent regime transitions throughout the sequence.

Results on Fixed-$K3$ dataset. Table 9 reports the complete results on the Fixed-$K3$ synthetic dataset. RegimeUnfold achieves the best overall performance in both regime inference and forecasting. Compared with Fixed-$K2$, the presence of an additional regime increases the difficulty of separating regime-specific dynamics, yet RegimeUnfold still maintains high Accuracy and F1 while achieving low forecasting errors. This indicates that the learned regime space remains effective as

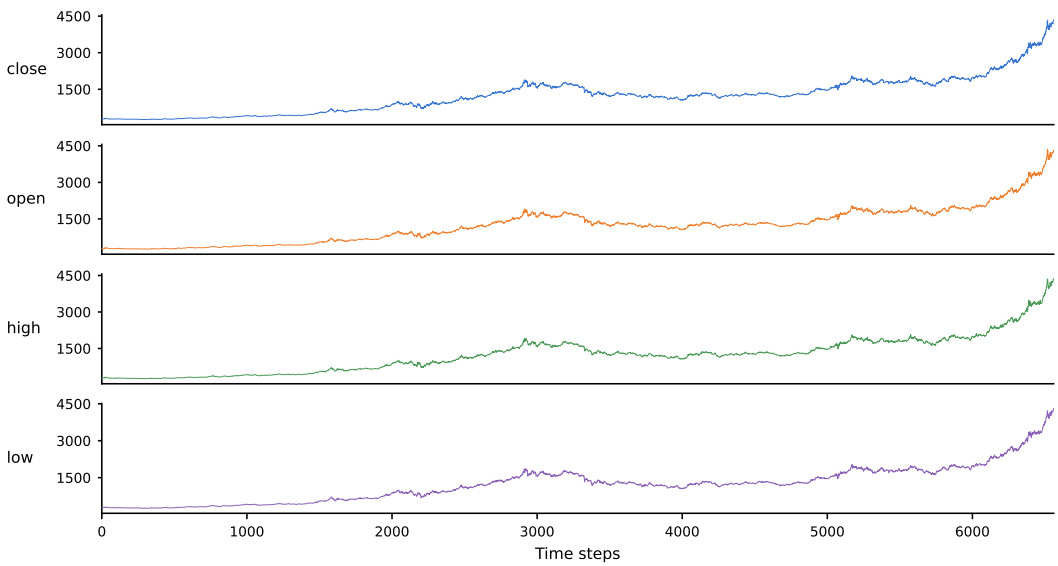


Figure 12: Gold futures price series with open, close, high, and low prices.

Table 7: Full results on the Fixed-$K2$ synthetic dataset.

Mode	Model	Accuracy (%)	F1 (%)	MAE	MSE
Non-adaptive	HMM	83.59	82.42	0.0735	0.01289
	HSMM	54.19	51.45	0.0569	0.00635
	HOHMM	82.24	81.12	0.0737	0.01296
	NHMM	70.45 ± 0.00	41.33 ± 0.00	0.0224 ± 0.0015	0.00127 ± 0.00013
	NCTRL	81.15 ± 13.33	69.34 ± 26.59	0.0212 ± 0.0081	0.00144 ± 0.00138
	Markovian-RNN	69.98 ± 0.81	42.54 ± 0.79	0.0124 ± 0.0008	0.00030 ± 0.00003
	DEN-HMM	71.55 ± 2.47	47.72 ± 14.29	0.0707 ± 0.0040	0.01084 ± 0.00142
	DRL-STAF	93.99 ± 1.97	93.11 ± 2.13	0.0099 ± 0.0002	0.00029 ± 0.00002
	iTransformer	-	-	0.0248 ± 0.0016	0.00128 ± 0.00009
	iTransformer (Large window)	-	-	0.0112 ± 0.0002	0.00026 ± 0.00001
	PatchTST	-	-	0.0241 ± 0.0007	0.00128 ± 0.00008
	PatchTST (Large window)	-	-	0.0109 ± 0.0001	0.00026 ± 0.00001
	TimeMixer	-	-	0.0278 ± 0.0045	0.00173 ± 0.00030
	TimeMixer (Large window)	-	-	0.0135 ± 0.0029	0.00095 ± 0.00133
Adaptive	BR-iHMM	96.58 ± 0.09	95.92 ± 0.10	0.0094 ± 0.0009	0.00028 ± 0.00005
	NHMM	70.45 ± 0.00	41.33 ± 0.00	0.0171 ± 0.0010	0.00073 ± 0.00010
	NCTRL	71.64 ± 22.45	63.75 ± 28.98	0.0160 ± 0.0043	0.00068 ± 0.00037
	Markovian-RNN	70.08 ± 0.61	42.74 ± 0.94	0.0103 ± 0.0002	0.00023 ± 0.00001
	DEN-HMM	68.61 ± 4.11	44.50 ± 7.09	0.0364 ± 0.0023	0.00304 ± 0.00038
	iTransformer	-	-	0.0201 ± 0.0010	0.00087 ± 0.00006
	iTransformer (Large window)	-	-	0.0112 ± 0.0002	0.00026 ± 0.00001
	PatchTST	-	-	0.0198 ± 0.0004	0.00087 ± 0.00003
	PatchTST (Large window)	-	-	0.0114 ± 0.0002	0.00028 ± 0.00001
	TimeMixer	-	-	0.0222 ± 0.0023	0.00112 ± 0.00013
	TimeMixer (Large window)	-	-	0.0123 ± 0.0008	0.00036 ± 0.00014
	RegimeUnfold	**97.85 ± 2.22**	**97.68 ± 2.08**	**0.0093 ± 0.0002**	**0.00022 ± 0.00001**

the number of latent regimes increases. Figure 15 further shows that RegimeUnfold closely tracks the observations and accurately recovers the transitions among the three regimes.

Results on Fixed-$K5$ dataset. Table 10 reports the complete results on the Fixed-$K5$ synthetic dataset. As the number of latent regimes increases from three to five, regime separation becomes substantially more challenging. RegimeUnfold nevertheless achieves the best overall performance, with clear advantages in regime inference and competitive forecasting accuracy. Although its Accuracy and F1 decrease compared with Fixed-$K2$ and Fixed-$K3$, the forecasting errors remain low, indicating that the learned regime-specific predictors continue to capture the underlying predictive dynamics even under a more complex regime space. Figure 16 further shows that RegimeUnfold can track the observations and recover the major regime transitions across the five-regime sequence.

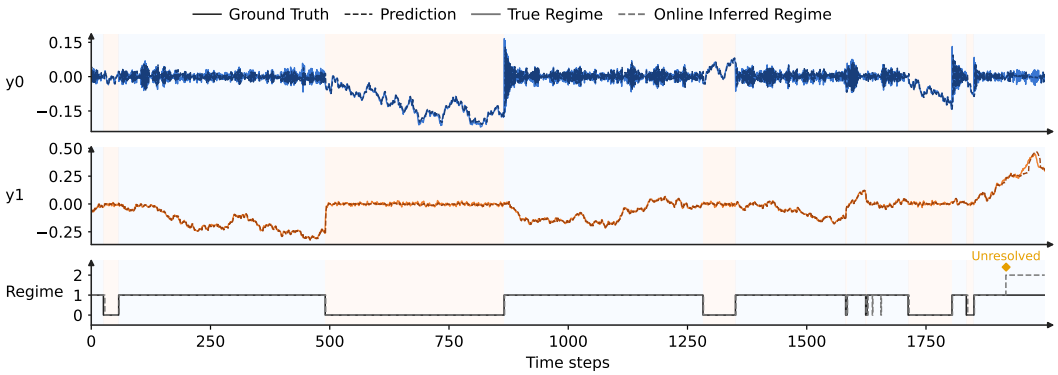


Figure 13: Results on the Fixed-$K2$ synthetic dataset.

Table 8: Full results on the Fast-$K2$ synthetic dataset.

Mode	Model	Accuracy (%)	F1 (%)	MAE	MSE
Non-adaptive	HMM	63.14	55.52	0.0316	0.00221
	HSMM	51.86	51.62	0.0313	0.00194
	HOHMM	54.48	54.40	0.0278	0.00160
	NHMM	58.61 ± 0.00	36.95 ± 0.00	0.0240 ± 0.0005	0.00151 ± 0.00011
	NCTRL	75.43 ± 21.19	75.01 ± 21.28	0.0188 ± 0.0069	0.00102 ± 0.00060
	Markovian-RNN	58.53 ± 0.99	38.66 ± 2.18	0.0132 ± 0.0000	0.00050 ± 0.00000
	DEN-HMM	59.46 ± 1.89	39.50 ± 5.69	0.0331 ± 0.0015	0.00238 ± 0.00012
	DRL-STAF	88.58 ± 0.65	88.10 ± 0.77	0.0110 ± 0.0001	0.00048 ± 0.00000
	iTransformer	-	-	0.0267 ± 0.0053	0.00170 ± 0.00045
	iTransformer (Large window)	-	-	0.0136 ± 0.0002	0.00052 ± 0.00001
	PatchTST	-	-	0.0243 ± 0.0004	0.00150 ± 0.00004
	PatchTST (Large window)	-	-	0.0136 ± 0.0001	0.00053 ± 0.00001
	TimeMixer	-	-	0.0249 ± 0.0008	0.00154 ± 0.00002
	TimeMixer (Large window)	-	-	0.0139 ± 0.0003	0.00052 ± 0.00001
Adaptive	BR-iHMM	85.74 ± 0.94	85.29 ± 1.00	0.0134 ± 0.0000	0.00086 ± 0.00001
	NHMM	57.62 ± 2.21	40.30 ± 7.48	0.0239 ± 0.0009	0.00154 ± 0.00010
	NCTRL	75.81 ± 18.19	75.60 ± 18.10	0.0179 ± 0.0048	0.00100 ± 0.00048
	Markovian-RNN	58.67 ± 0.69	38.90 ± 2.37	0.0129 ± 0.0001	0.00050 ± 0.00000
	DEN-HMM	58.79 ± 0.41	41.00 ± 9.06	0.0331 ± 0.0012	0.00242 ± 0.00009
	iTransformer	-	-	0.0255 ± 0.0033	0.00156 ± 0.00028
	iTransformer (Large window)	-	-	0.0143 ± 0.0002	0.00057 ± 0.00001
	PatchTST	-	-	0.0234 ± 0.0003	0.00138 ± 0.00002
	PatchTST (Large window)	-	-	0.0140 ± 0.0002	0.00058 ± 0.00001
	TimeMixer	-	-	0.0244 ± 0.0006	0.00147 ± 0.00003
	TimeMixer (Large window)	-	-	0.0146 ± 0.0003	0.00057 ± 0.00001
	RegimeUnfold	**93.01 ± 0.11**	**92.83 ± 0.12**	**0.0106 ± 0.0000**	**0.00046 ± 0.00000**

Results on Progressive dataset. Table 11 reports the complete results on the Progressive synthetic dataset, where previously unseen predictive dynamics gradually expand the available regime space. RegimeUnfold achieves a clear advantage in regime inference while maintaining strong forecasting performance, demonstrating its ability to adapt when the current regime space becomes insufficient. Figure 17 further illustrates the online evolution process. When the new regime first appears, RegimeUnfold does not immediately expand the regime space because persistent evidence must first be accumulated, resulting in a noticeable delay in the "NEW" decision. After the new regime is incorporated, however, the same dynamics are recognized much more promptly when they reappear later, and the corresponding regime is stably reused for forecasting. This behavior reflects the trade-off between conservative regime creation and reliable subsequent regime reuse.

E.2 Results on Real-World Datasets

Results on SMD dataset. Table 12 reports the complete results on SMD, which provides ground-truth anomaly labels for evaluating regime inference in addition to forecasting performance. RegimeUnfold achieves the best overall regime inference performance, with the highest Accuracy and F1, while also maintaining strong forecasting accuracy. Since the anomaly labels are used only for evaluation and are not involved in model training or online regime learning, these results

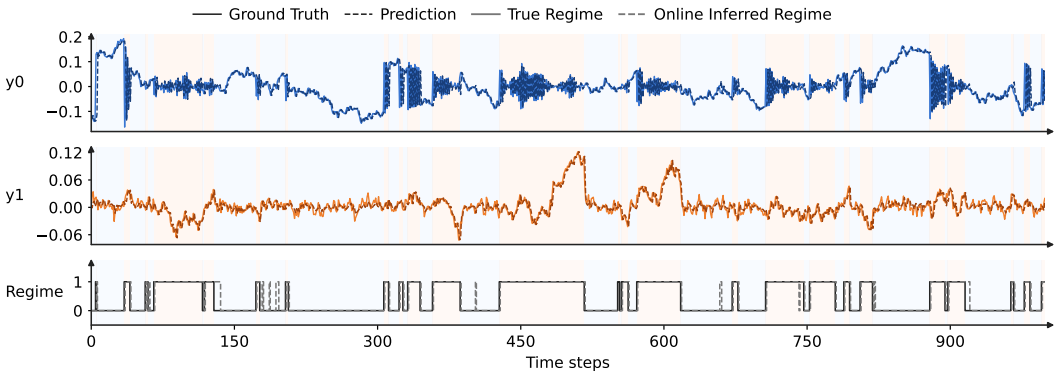


Figure 14: Results on the Fast-$K2$ synthetic dataset.

Table 9: Full results on the Fixed-$K3$ synthetic dataset.

Mode	Model	Accuracy (%)	F1 (%)	MAE	MSE
Non-adaptive	HMM	72.10	71.53	0.0395	0.00497
	HSMM	61.11	60.62	0.0385	0.00413
	HOHMM	76.77	76.07	0.0429	0.00591
	NHMM	44.56 ± 0.00	20.55 ± 0.00	0.0206 ± 0.0007	0.00105 ± 0.00003
	NCTRL	63.08 ± 5.52	55.36 ± 3.55	0.0280 ± 0.0042	0.00197 ± 0.00065
	Markovian-RNN	44.54 ± 0.29	23.90 ± 0.69	0.0174 ± 0.0012	0.00069 ± 0.00008
	DEN-HMM	46.27 ± 5.55	32.58 ± 7.85	0.0392 ± 0.0072	0.00472 ± 0.00197
	DRL-STAF	66.31 ± 1.13	58.05 ± 5.42	0.0160 ± 0.0005	0.00086 ± 0.00008
	iTransformer	-	-	0.0205 ± 0.0011	0.00105 ± 0.00010
	iTransformer (Large window)	-	-	0.0121 ± 0.0002	0.00033 ± 0.00001
	PatchTST	-	-	0.0205 ± 0.0012	0.00105 ± 0.00016
	PatchTST (Large window)	-	-	0.0120 ± 0.0001	0.00035 ± 0.00001
	TimeMixer	-	-	0.0242 ± 0.0005	0.00154 ± 0.00009
	TimeMixer (Large window)	-	-	0.0125 ± 0.0002	0.00035 ± 0.00001
Adaptive	BR-iHMM	73.64 ± 5.13	59.07 ± 11.63	0.0101 ± 0.0003	0.00041 ± 0.00001
	NHMM	44.56 ± 0.00	20.55 ± 0.00	0.0214 ± 0.0015	0.00120 ± 0.00018
	NCTRL	52.47 ± 5.22	45.29 ± 8.41	0.0231 ± 0.0012	0.00146 ± 0.00017
	Markovian-RNN	44.61 ± 0.07	20.98 ± 0.12	0.0126 ± 0.0004	0.00037 ± 0.00004
	DEN-HMM	43.37 ± 3.73	31.50 ± 7.34	0.0323 ± 0.0024	0.00262 ± 0.00044
	iTransformer	-	-	0.0194 ± 0.0010	0.00097 ± 0.00007
	iTransformer (Large window)	-	-	0.0121 ± 0.0002	0.00035 ± 0.00001
	PatchTST	-	-	0.0197 ± 0.0003	0.00099 ± 0.00003
	PatchTST (Large window)	-	-	0.0125 ± 0.0002	0.00038 ± 0.00002
	TimeMixer	-	-	0.0234 ± 0.0009	0.00140 ± 0.00012
	TimeMixer (Large window)	-	-	0.0126 ± 0.0001	0.00037 ± 0.00001
	RegimeUnfold	**96.31 ± 0.23**	**95.88 ± 0.28**	**0.0093 ± 0.0001**	**0.00026 ± 0.00001**

indicate that the inferred latent regimes are meaningfully aligned with the underlying normal and anomalous states. Figure 18 further presents the forecasting and regime inference results, showing that RegimeUnfold closely tracks the observed sequence while recovering the major transitions between the reference states.

Results on Exchange, Traffic, and GOLD datasets. Table 13 reports the complete forecasting results on the Exchange, Traffic, and GOLD datasets. Across these three datasets, RegimeUnfold achieves consistently strong forecasting performance under diverse real-world dynamics. It obtains the lowest MSE on Exchange, the lowest MAE on Traffic, and the lowest MAE and MSE on GOLD, while remaining competitive on the other metrics. These results show that the learned latent regime structure and online adaptation remain useful even when explicit regime annotations are unavailable. The consistent performance across foreign-exchange rates, traffic speeds, and gold futures prices further demonstrates the applicability of RegimeUnfold to heterogeneous real-world sequential data.

Figures 19-21 further present the forecasting results of the top seven models under the adaptive setting, with different background colors indicating the regimes inferred by RegimeUnfold. The predictions of RegimeUnfold closely follow the ground-truth observations across different inferred regimes. In contrast, some adaptive baselines based on periodic sliding-window fine-tuning ex-

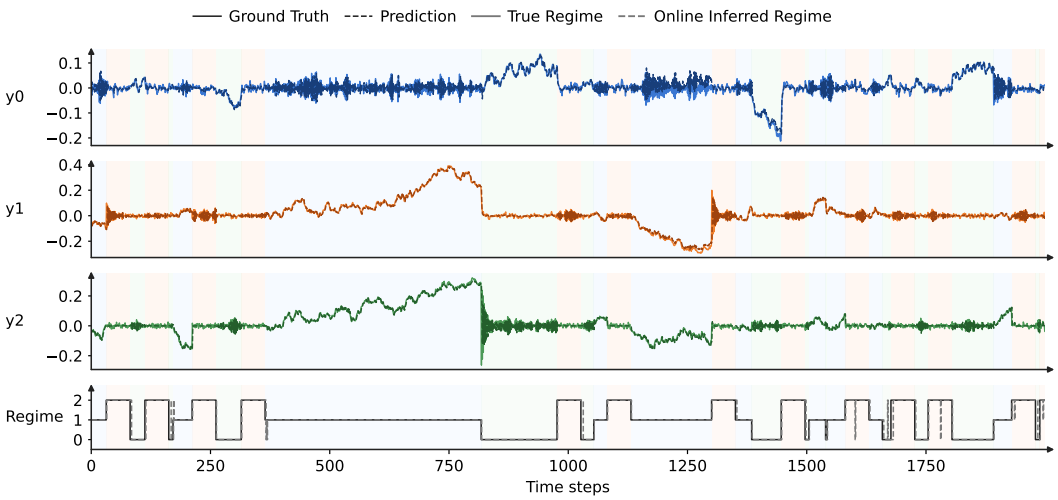


Figure 15: Results on the Fixed-$K3$ synthetic dataset.

Table 10: Full results on the Fixed-$K5$ synthetic dataset.

Mode	Model	Accuracy (%)	F1 (%)	MAE	MSE
Non-adaptive	HMM	60.21	49.91	0.0191	0.00099
	HSMM	66.88	64.80	0.0238	0.00180
	HOHMM	60.36	58.84	0.0245	0.00233
	NHMM	40.34 ± 0.00	11.50 ± 0.00	0.0157 ± 0.0003	0.00057 ± 0.00001
	NCTRL	63.21 ± 5.18	50.09 ± 5.23	0.0202 ± 0.0038	0.00112 ± 0.00056
	Markovian-RNN	39.88 ± 0.46	13.37 ± 0.77	0.0134 ± 0.0001	0.00038 ± 0.00001
	DEN-HMM	48.65 ± 9.05	24.38 ± 12.83	0.0253 ± 0.0036	0.00166 ± 0.00042
	DRL-STAF	42.53 ± 7.18	22.51 ± 2.69	0.0180 ± 0.0005	0.00078 ± 0.00005
	iTransformer	-	-	0.0168 ± 0.0023	0.00064 ± 0.00014
	iTransformer (Large window)	-	-	0.0108 ± 0.0001	0.00022 ± 0.00000
	PatchTST	-	-	0.0170 ± 0.0021	0.00066 ± 0.00015
	PatchTST (Large window)	-	-	0.0106 ± 0.0004	0.00022 ± 0.00001
	TimeMixer	-	-	0.0171 ± 0.0006	0.00065 ± 0.00003
	TimeMixer (Large window)	-	-	0.0113 ± 0.0001	0.00024 ± 0.00001
Adaptive	BR-iHMM	67.88 ± 2.08	36.32 ± 7.40	0.0111 ± 0.0008	0.00034 ± 0.00004
	NHMM	40.34 ± 0.00	11.50 ± 0.00	0.0161 ± 0.0001	0.00062 ± 0.00002
	NCTRL	48.60 ± 5.34	34.71 ± 4.85	0.0179 ± 0.0014	0.00078 ± 0.00016
	Markovian-RNN	40.37 ± 0.06	11.98 ± 0.54	0.0130 ± 0.0001	0.00036 ± 0.00001
	DEN-HMM	41.42 ± 2.65	23.21 ± 4.19	0.0220 ± 0.0011	0.00129 ± 0.00016
	iTransformer	-	-	0.0155 ± 0.0010	0.00056 ± 0.00006
	iTransformer (Large window)	-	-	0.0112 ± 0.0000	0.00024 ± 0.00000
	PatchTST	-	-	0.0155 ± 0.0009	0.00057 ± 0.00007
	PatchTST (Large window)	-	-	0.0112 ± 0.0003	0.00024 ± 0.00001
	TimeMixer	-	-	0.0161 ± 0.0002	0.00058 ± 0.00002
	TimeMixer (Large window)	-	-	0.0117 ± 0.0002	0.00026 ± 0.00001
	RegimeUnfold	**88.68 ± 7.49**	**76.42 ± 16.46**	**0.0091 ± 0.0008**	**0.00018 ± 0.00004**

hibit visible step-like discontinuities or abrupt level shifts during online adaptation. RegimeUnfold produces smoother and more consistent predictions while adapting to changing dynamics, further demonstrating its strong forecasting performance on heterogeneous real-world sequences.

E.3 Ablation Study

Results on synthetic datasets. Tables 14-18 examine the effects of explicit regime modeling and regime-space evolution. Removing regime modeling generally leads to substantially higher forecasting errors, while allowing the no-regime model to adapt online reduces part of this gap, indicating that online parameter adaptation itself is beneficial but cannot fully replace regime-specific modeling. On the fixed-regime datasets, RegimeUnfold and RegimeUnfold w/o Evolution achieve similar performance, with small differences in regime inference and forecasting because no new regimes need to be introduced. In contrast, the gap becomes substantial on the Progressive dataset: without regime-space evolution, regime inference deteriorates sharply and forecasting errors increase,

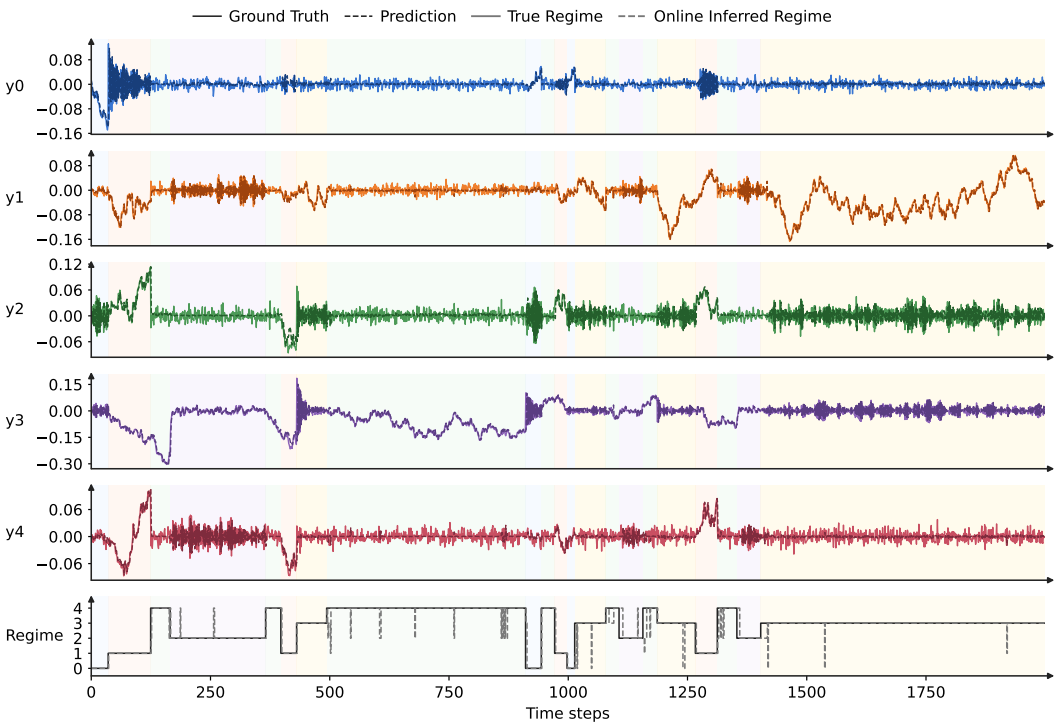


Figure 16: Results on the Fixed-$K5$ synthetic dataset.

Table 11: Full results on the Progressive synthetic dataset.

Mode	Model	Accuracy (%)	F1 (%)	MAE	MSE
Non-adaptive	HMM	51.81	35.31	0.0590	0.00577
	HSMM	51.21	34.81	0.0602	0.00601
	HOHMM	51.81	35.31	0.0590	0.00577
	NHMM	30.47 ± 0.00	11.68 ± 0.00	0.0237 ± 0.0006	0.00141 ± 0.00005
	NCTRL	41.25 ± 6.94	26.47 ± 5.27	0.0342 ± 0.0056	0.00284 ± 0.00066
	Markovian-RNN	30.84 ± 0.32	12.96 ± 0.76	0.0200 ± 0.0007	0.00113 ± 0.00006
	DEN-HMM	66.27 ± 20.02	52.52 ± 22.83	0.0471 ± 0.0119	0.00419 ± 0.00162
	DRL-STAF	58.60 ± 1.64	54.81 ± 3.76	0.0330 ± 0.0005	0.00278 ± 0.00007
	iTransformer	-	-	0.0280 ± 0.0025	0.00176 ± 0.00033
	iTransformer (Large window)	-	-	0.0156 ± 0.0005	**0.00077 ± 0.00005**
	PatchTST	-	-	0.0214 ± 0.0008	0.00123 ± 0.00006
	PatchTST (Large window)	-	-	0.0161 ± 0.0003	0.00087 ± 0.00002
	TimeMixer	-	-	0.0401 ± 0.0049	0.00367 ± 0.00085
	TimeMixer (Large window)	-	-	0.0162 ± 0.0009	0.00083 ± 0.00006
Adaptive	BR-iHMM	45.53 ± 1.27	30.41 ± 0.99	0.0215 ± 0.0003	0.00167 ± 0.00003
	NHMM	30.47 ± 0.00	11.68 ± 0.00	0.0260 ± 0.0011	0.00187 ± 0.00016
	NCTRL	35.24 ± 4.24	33.30 ± 4.78	0.0403 ± 0.0102	0.00378 ± 0.00137
	Markovian-RNN	30.88 ± 0.69	13.09 ± 2.58	0.0171 ± 0.0004	0.00096 ± 0.00004
	DEN-HMM	50.33 ± 7.50	45.67 ± 12.24	0.0376 ± 0.0128	0.00309 ± 0.00194
	iTransformer	-	-	0.0222 ± 0.0007	0.00125 ± 0.00006
	iTransformer (Large window)	-	-	0.0163 ± 0.0002	0.00081 ± 0.00002
	PatchTST	-	-	0.0219 ± 0.0008	0.00126 ± 0.00006
	PatchTST (Large window)	-	-	0.0172 ± 0.0003	0.00087 ± 0.00002
	TimeMixer	-	-	0.0249 ± 0.0020	0.00149 ± 0.00020
	TimeMixer (Large window)	-	-	0.0164 ± 0.0005	0.00082 ± 0.00003
	RegimeUnfold	**91.49 ± 0.61**	**91.19 ± 0.77**	**0.0134 ± 0.0000**	0.00081 ± 0.00001

whereas the full model maintains accurate forecasting as new dynamics emerge. These results show that explicit regime modeling improves the representation of heterogeneous predictive dynamics, while regime-space evolution becomes essential when the current regime space is no longer sufficient.

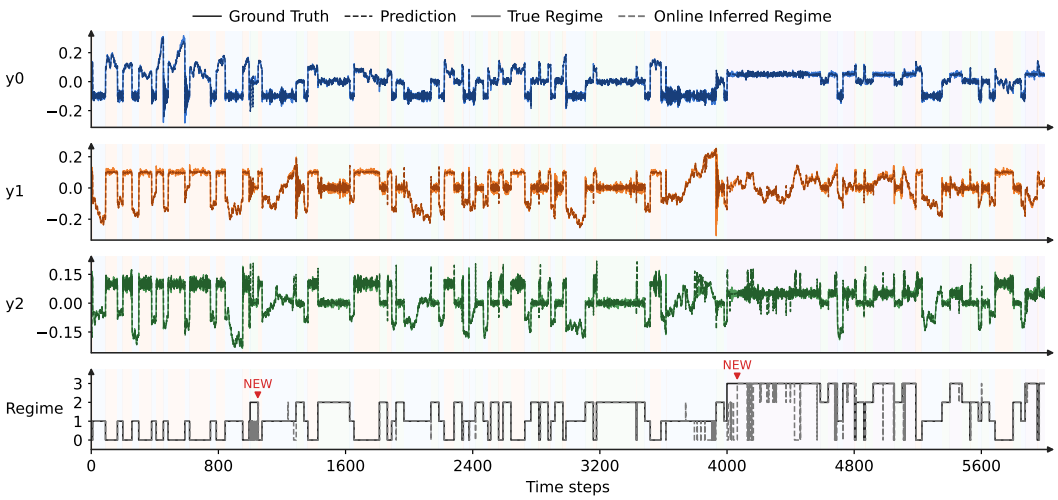


Figure 17: Results on the Progressive synthetic dataset.

Table 12: Full results on the SMD dataset.

Mode	Model	Accuracy (%)	F1 (%)	MAE	MSE
Non-adaptive	HMM	54.16	48.42	0.1341	0.02883
	HSMM	55.60	50.25	0.1327	0.02831
	HOHMM	54.81	49.07	0.1320	0.02792
	NHMM	70.66 ± 0.00	41.40 ± 0.00	0.0221 ± 0.0009	0.00109 ± 0.00005
	NCTRL	80.72 ± 11.62	73.65 ± 16.60	0.0308 ± 0.0034	0.00257 ± 0.00064
	Markovian-RNN	70.36 ± 0.51	43.96 ± 0.49	0.0193 ± 0.0003	0.00094 ± 0.00001
	DEN-HMM	70.80 ± 0.32	42.09 ± 1.53	0.1501 ± 0.0123	0.03730 ± 0.00350
	DRL-STAF	85.63 ± 0.00	81.60 ± 0.00	0.0218 ± 0.0009	0.00149 ± 0.00023
	iTransformer	-	-	0.0241 ± 0.0006	0.00127 ± 0.00016
	iTransformer (Large window)	-	-	0.0216 ± 0.0009	0.00107 ± 0.00011
	PatchTST	-	-	0.0212 ± 0.0005	0.00103 ± 0.00003
	PatchTST (Large window)	-	-	0.0197 ± 0.0003	0.00095 ± 0.00001
	TimeMixer	-	-	0.0261 ± 0.0008	0.00183 ± 0.00039
	TimeMixer (Large window)	-	-	0.0216 ± 0.0008	0.00110 ± 0.00009
Adaptive	BR-iHMM	70.66 ± 0.00	41.40 ± 0.00	**0.0183 ± 0.0000**	0.00096 ± 0.00000
	NHMM	75.55 ± 5.95	63.23 ± 20.24	0.0247 ± 0.0010	0.00144 ± 0.00017
	NCTRL	71.06 ± 11.04	63.01 ± 15.30	0.0256 ± 0.0006	0.00170 ± 0.00009
	Markovian-RNN	70.66 ± 0.13	42.20 ± 0.29	0.0185 ± 0.0001	0.00095 ± 0.00001
	DEN-HMM	64.99 ± 8.38	47.42 ± 8.83	0.0615 ± 0.0038	0.01164 ± 0.00165
	iTransformer	-	-	0.0272 ± 0.0027	0.00158 ± 0.00032
	iTransformer (Large window)	-	-	0.0264 ± 0.0028	0.00156 ± 0.00038
	PatchTST	-	-	0.0280 ± 0.0033	0.00183 ± 0.00042
	PatchTST (Large window)	-	-	0.0240 ± 0.0007	0.00131 ± 0.00008
	TimeMixer	-	-	0.0341 ± 0.0045	0.00253 ± 0.00056
	TimeMixer (Large window)	-	-	0.0250 ± 0.0011	0.00137 ± 0.00009
	RegimeUnfold	**89.65 ± 6.75**	**85.82 ± 16.04**	**0.0183 ± 0.0001**	**0.00092 ± 0.00000**

Results on real-world datasets. Tables 19 and 20 further evaluate the same components on real-world data. On SMD, removing regime-space evolution reduces regime inference performance, whereas the full model better aligns the inferred regimes with the ground-truth anomaly states while maintaining low forecasting errors. On Exchange, Traffic, and GOLD, RegimeUnfold generally achieves the strongest forecasting performance among the ablated variants, although the effect of simple online adaptation without regime modeling varies across datasets. This suggests that repeatedly fine-tuning a single predictor is not sufficient to consistently handle heterogeneous and evolving dynamics, while explicit regime modeling together with regime-space evolution provides a more robust mechanism for real-world forecasting.

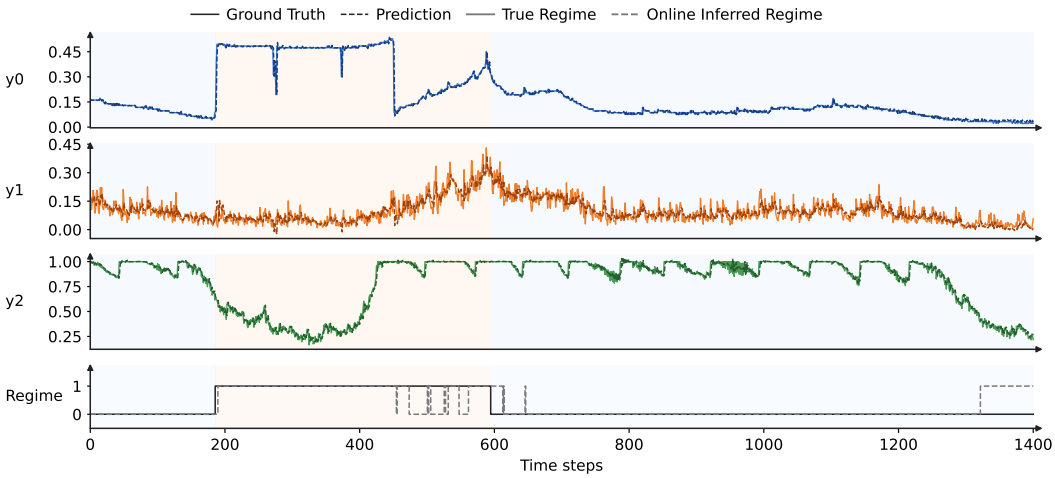


Figure 18: Results on the SMD dataset.

Table 13: Full results on the Exchange, Traffic, and GOLD datasets.

Mode	Model	Exchange		Traffic		GOLD	
		MAE	MSE	MAE	MSE	MAE	MSE
Non-adaptive	HMM	7.9617	248.7797	5.4173	67.6396	984.65	1381161.90
	HSMM	7.9634	248.8891	5.4124	67.3115	984.78	1381436.92
	HOHMM	7.9624	248.8203	5.4171	67.6296	984.64	1381160.69
	NHMM	1.8489 ± 0.6785	17.4428 ± 9.8664	1.5881 ± 0.0241	6.7800 ± 0.0850	250.44 ± 71.65	248913.75 ± 122501.54
	NCTRL	3.5346 ± 1.4861	61.5884 ± 50.5065	2.1324 ± 0.1630	10.9013 ± 1.7096	290.13 ± 131.54	284846.18 ± 273530.73
	Markovian-RNN	1.6795 ± 0.1360	12.3895 ± 1.8464	1.9590 ± 0.1602	7.9649 ± 1.3491	197.10 ± 72.81	105203.85 ± 60373.18
	DEN-HMM	10.8710 ± 1.4300	427.8633 ± 90.5533	7.6843 ± 3.4121	107.7766 ± 75.2487	1206.41 ± 164.87	1891706.24 ± 365890.87
	DRL-STAF	2.1519 ± 0.0428	19.6272 ± 0.7645	1.5406 ± 0.0071	6.6540 ± 0.0927	314.72 ± 3.75	233775.77 ± 3285.98
	iTransformer	0.6834 ± 0.4396	2.9078 ± 3.7178	1.6371 ± 0.0145	5.9855 ± 0.0746	275.24 ± 119.30	361721.66 ± 336993.15
	iTransformer (Large window)	1.1897 ± 1.5632	14.9382 ± 30.6615	1.5976 ± 0.0170	5.7274 ± 0.0745	281.66 ± 126.32	401899.70 ± 363340.91
	PatchTST	0.3539 ± 0.1249	0.6805 ± 0.5185	1.6144 ± 0.0191	5.8363 ± 0.1511	181.02 ± 99.66	128705.79 ± 105857.02
	PatchTST (Large window)	0.3638 ± 0.0685	0.6880 ± 0.2603	1.5780 ± 0.0171	5.5308 ± 0.1124	154.53 ± 86.23	120046.07 ± 116938.73
	TimeMixer	3.8829 ± 0.9007	69.0204 ± 31.8386	1.8315 ± 0.0570	7.6385 ± 0.5374	499.04 ± 68.09	928360.28 ± 329341.17
	TimeMixer (Large window)	2.1722 ± 0.6952	22.8618 ± 13.7627	1.6153 ± 0.0110	5.9445 ± 0.1072	268.03 ± 205.63	415485.84 ± 673255.49
Adaptive	BR-iHMM	**0.2119 ± 0.0001**	0.2570 ± 0.0002	1.5522 ± 0.0002	**5.5274 ± 0.0030**	12.70 ± 0.00	392.96 ± 1.21
	NHMM	0.4664 ± 0.0725	1.2472 ± 0.3592	2.1291 ± 0.0255	11.5472 ± 0.2891	40.24 ± 5.96	4624.38 ± 831.64
	NCTRL	0.7139 ± 0.3933	3.0579 ± 2.5407	2.7940 ± 0.4037	20.1602 ± 6.6182	79.41 ± 116.13	28148.52 ± 58097.00
	Markovian-RNN	0.2642 ± 0.0055	0.3730 ± 0.0134	1.5429 ± 0.0089	5.8620 ± 0.0655	16.10 ± 0.94	631.43 ± 87.34
	DEN-HMM	1.3514 ± 0.2927	9.2999 ± 3.8832	8.2739 ± 2.7221	137.1624 ± 74.1098	95.83 ± 17.65	22945.68 ± 6925.74
	iTransformer	0.9331 ± 0.2016	4.4694 ± 1.7446	1.8773 ± 0.0349	7.8555 ± 0.3530	56.83 ± 6.00	7210.17 ± 1192.46
	iTransformer (Large window)	0.6034 ± 0.2075	1.9530 ± 1.5034	1.8764 ± 0.0391	7.7789 ± 0.2614	56.26 ± 4.39	6844.89 ± 715.55
	PatchTST	0.5095 ± 0.2338	1.6576 ± 1.5329	1.8561 ± 0.0793	7.6720 ± 0.5229	45.11 ± 5.55	4656.06 ± 1135.37
	PatchTST (Large window)	0.6878 ± 0.2262	2.6341 ± 1.3751	1.9597 ± 0.0631	8.0918 ± 0.4490	58.88 ± 7.63	6836.05 ± 1459.11
	TimeMixer	1.1047 ± 0.1588	5.6313 ± 1.4750	2.1425 ± 0.1563	11.1989 ± 1.5435	60.21 ± 14.73	9069.95 ± 4319.51
	TimeMixer (Large window)	0.8797 ± 0.1818	3.7881 ± 1.4146	2.0208 ± 0.1305	8.8330 ± 1.0653	65.81 ± 13.60	10241.27 ± 4317.93
	RegimeUnfold	0.2146 ± 0.0030	**0.2561 ± 0.0026**	**1.5323 ± 0.0420**	5.7416 ± 0.1621	**11.82 ± 0.09**	**295.19 ± 6.31**

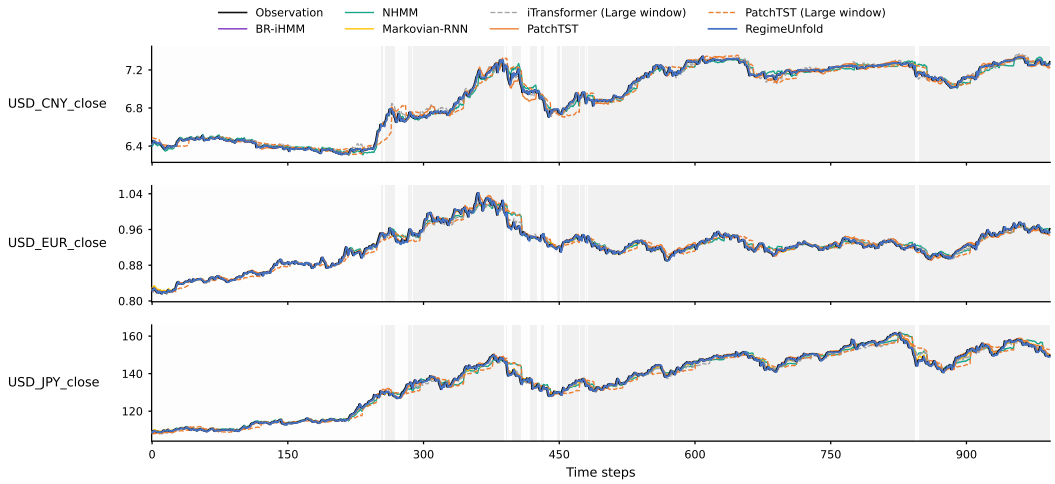


Figure 19: Results on the Exchange dataset.

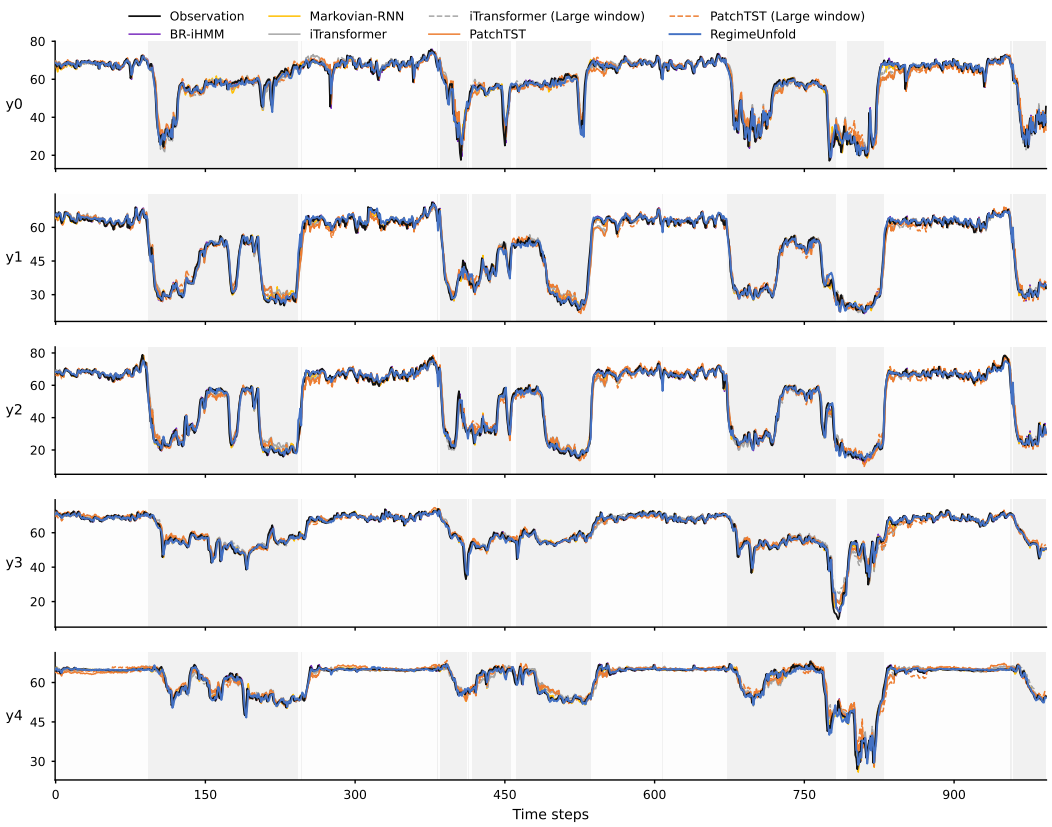


Figure 20: Results on the Traffic dataset.

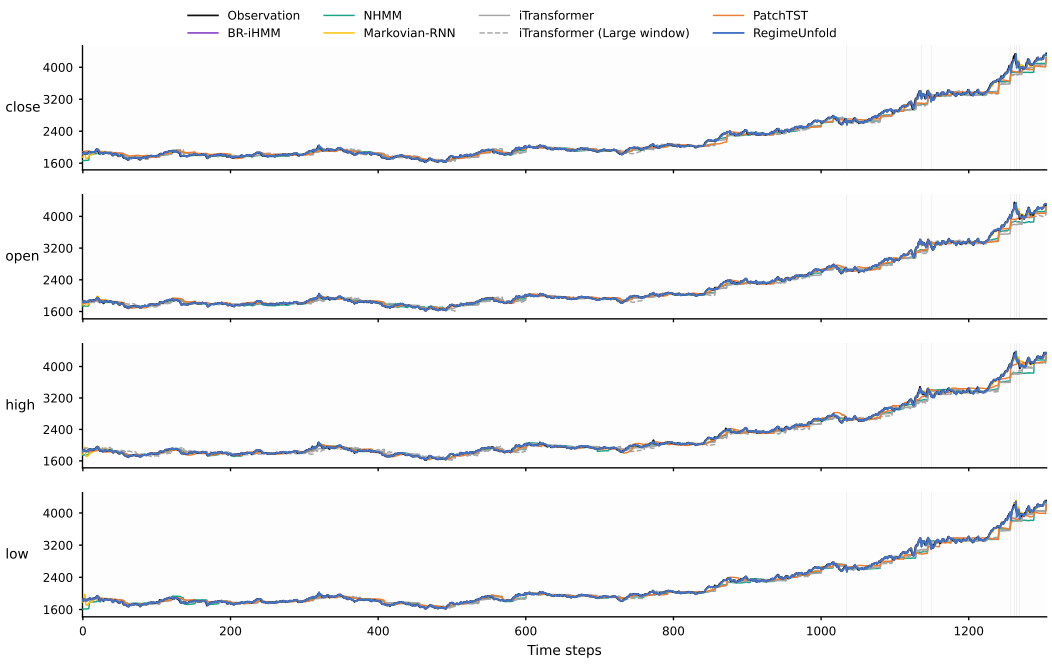


Figure 21: Results on the GOLD dataset.

Table 14: Ablation results on the Fixed-$K2$ synthetic dataset.

Model	Accuracy (%)	F1 (%)	MAE	MSE
RegimeUnfold w/o Regime (Non-adaptive)	-	-	0.0236 ± 0.0012	0.00131 ± 0.00010
RegimeUnfold w/o Regime (Adaptive)	-	-	0.0184 ± 0.0005	0.00079 ± 0.00003
RegimeUnfold w/o Evolution	97.75 ± 2.47	97.58 ± 2.33	0.0100 ± 0.0005	0.00034 ± 0.00008
RegimeUnfold	**97.85 ± 2.22**	**97.68 ± 2.08**	**0.0093 ± 0.0002**	**0.00022 ± 0.00001**

Table 15: Ablation results on the Fast-$K2$ synthetic dataset.

Model	Accuracy (%)	F1 (%)	MAE	MSE
RegimeUnfold w/o Regime (Non-adaptive)	-	-	0.0241 ± 0.0005	0.00145 ± 0.00005
RegimeUnfold w/o Regime (Adaptive)	-	-	0.0230 ± 0.0004	0.00137 ± 0.00002
RegimeUnfold w/o Evolution	**93.01 ± 0.11**	**92.83 ± 0.12**	**0.0106 ± 0.0000**	**0.00046 ± 0.00000**
RegimeUnfold	**93.01 ± 0.11**	**92.83 ± 0.12**	**0.0106 ± 0.0000**	**0.00046 ± 0.00000**

Table 16: Ablation results on the Fixed-$K3$ synthetic dataset.

Model	Accuracy (%)	F1 (%)	MAE	MSE
RegimeUnfold w/o Regime (Non-adaptive)	-	-	0.0186 ± 0.0008	0.00095 ± 0.00008
RegimeUnfold w/o Regime (Adaptive)	-	-	0.0178 ± 0.0007	0.00087 ± 0.00005
RegimeUnfold w/o Evolution	**96.33 ± 0.25**	**95.90 ± 0.31**	0.0094 ± 0.0002	0.00026 ± 0.00002
RegimeUnfold	96.31 ± 0.23	95.88 ± 0.28	**0.0093 ± 0.0001**	**0.00026 ± 0.00001**

Table 17: Ablation results on the Fixed-$K5$ synthetic dataset.

Model	Accuracy (%)	F1 (%)	MAE	MSE
RegimeUnfold w/o Regime (Non-adaptive)	-	-	0.0153 ± 0.0002	0.00056 ± 0.00001
RegimeUnfold w/o Regime (Adaptive)	-	-	0.0146 ± 0.0001	0.00052 ± 0.00001
RegimeUnfold w/o Evolution	**88.72 ± 7.50**	**76.51 ± 16.56**	0.0091 ± 0.0008	**0.00018 ± 0.00004**
RegimeUnfold	88.68 ± 7.49	76.42 ± 16.46	**0.0091 ± 0.0008**	0.00018 ± 0.00004

Table 18: Ablation results on the Progressive synthetic dataset.

Model	Accuracy (%)	F1 (%)	MAE	MSE
RegimeUnfold w/o Regime (Non-adaptive)	-	-	0.0304 ± 0.0049	0.00229 ± 0.00060
RegimeUnfold w/o Regime (Adaptive)	-	-	0.0215 ± 0.0017	0.00133 ± 0.00018
RegimeUnfold w/o Evolution	51.56 ± 1.16	34.08 ± 1.33	0.0426 ± 0.0056	0.00508 ± 0.00123
RegimeUnfold	**91.49 ± 0.61**	**91.19 ± 0.77**	**0.0134 ± 0.0000**	**0.00081 ± 0.00001**

Table 19: Ablation results on the SMD dataset.

Model	Accuracy (%)	F1 (%)	MAE	MSE
RegimeUnfold w/o Regime (Non-adaptive)	-	-	0.0210 ± 0.0007	0.00108 ± 0.00008
RegimeUnfold w/o Regime (Adaptive)	-	-	0.0204 ± 0.0003	0.00100 ± 0.00002
RegimeUnfold w/o Evolution	74.52 ± 7.72	59.50 ± 18.70	0.0185 ± 0.0001	**0.00092 ± 0.00000**
RegimeUnfold	**89.65 ± 6.75**	**85.82 ± 16.04**	**0.0183 ± 0.0001**	0.00092 ± 0.00000

Table 20: Ablation results on the Exchange, Traffic, and GOLD datasets.

Model	Exchange		Traffic		GOLD	
	MAE	MSE	MAE	MSE	MAE	MSE
RegimeUnfold w/o Regime (Non-adaptive)	1.6881 ± 1.4638	20.5619 ± 32.9221	1.5376 ± 0.0331	5.7863 ± 0.2276	270.58 ± 89.63	282086.10 ± 144168.39
RegimeUnfold w/o Regime (Adaptive)	0.4524 ± 0.1868	1.3015 ± 1.0486	1.5958 ± 0.0186	6.1337 ± 0.1429	41.21 ± 8.40	6459.46 ± 2393.38
RegimeUnfold w/o Evolution	0.2157 ± 0.0020	0.2578 ± 0.0016	1.5373 ± 0.0386	5.8021 ± 0.1442	12.85 ± 0.12	391.23 ± 8.07
RegimeUnfold	**0.2146 ± 0.0030**	**0.2561 ± 0.0026**	**1.5323 ± 0.0420**	**5.7416 ± 0.1621**	**11.82 ± 0.09**	**295.19 ± 6.31**