CoEvoST: A Dual Evolution Agent for Continual Spatiotemporal Forecasting

Anonymous authors
Paper under double-blind review

Abstract

Spatiotemporal forecasting systems operate on evolving streams with changing data distributions, graph structures, observed node sets, and external conditions, requiring continual predictor adaptation. Existing continual spatiotemporal forecasting methods primarily focus on how to adapt the predictor through specialized mechanisms, while higher-level decisions about whether, where, on what data, and by which mechanism to adapt are typically prescribed rather than learned from evolving states. This raises two challenges: forming an appropriate adaptation plan from heterogeneous diagnostic evidence, and improving adaptation decisions from limited execution feedback. We therefore propose **CoEvoST**, a dual evolution framework that couples fast predictor adaptation with slower evolution of an LLM adaptation agent. For **evidence-to-action reasoning**, CoEvoST starts from compact diagnostic summaries and performs multi-round evidence retrieval and plan refinement, producing an adaptation plan that specifies whether, where, on what data, and by which mechanism to update the predictor with sufficient diagnostic support. For **outcome-to-policy learning**, CoEvoST executes candidate plans from the same predictor checkpoint and constructs near-optimal preference sets from their realized outcomes, accumulating them as experience for online field-weighted DPO to improve the agent's decision policy. Experiments across four spatiotemporal streams demonstrate that CoEvoST improves forecasting performance over strong fixed and continual baselines while adapting its decisions to changing operating conditions.

1 Introduction

Spatiotemporal forecasting supports traffic, energy, environmental monitoring, and water-resource management. In deployment, however, observations rarely come from stationary datasets with fixed topology. Demand patterns drift, seasonal phases shift, sensors are added or removed, node connections evolve, extreme events occur, and external conditions alter the relationship between historical observations and future states. A predictor trained once may consequently deteriorate over time. Indiscriminate retraining is not an ideal alternative: frequent updates require resources, short recent windows may induce catastrophic forgetting, and an unsuitable adaptation mechanism can amplify rather than correct the effects of change.

Continual learning has developed various mechanisms for adapting forecasting models to non-stationary environments. TrafficStream uses replay and weight consolidation (Chen et al., 2021); STKEC and PECPM introduce influence- or pattern-based memory (Wang et al., 2023a;b); EAC employs prompt tuning (Chen & Liang, 2025); ST-LoRA uses node-adaptive low-rank updates (Ruan et al., 2025); STRAP retrieves spatiotemporal patterns (Zhang et al., 2025); and ST-TTC performs frequency-domain test-time calibration (Chen & Liang, 2026). These methods substantially advance continual spatiotemporal forecasting by designing specialized mechanisms for adapting the predictor. However, these methods primarily address how to adapt the predictor, while the higher-level decision of whether, where, on what data, and by which mechanism to adapt is typically prescribed rather than learned from the evolving state. Such fixed decisions may be unsuitable when different types and spatial extents of change require different adaptation behaviors.

We therefore formulate predictor adaptation as a state-dependent decision problem, in which an agent determines whether to update the predictor, which data to use, where adaptation should be applied, and which mechanism to invoke based on diagnostic evidence. Moreover, as operating conditions continue to evolve, the effectiveness of these decisions may also change, requiring the

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.

decision policy itself to improve from accumulated execution experience. Together, these requirements present two key challenges.

How to form an appropriate adaptation plan from heterogeneous diagnostic evidence? Similar increases in forecast error may arise from distribution drift, sensor changes, graph updates, environmental shifts, or random fluctuations, each of which may call for a different adaptation response. Forming an appropriate plan therefore requires identifying the nature and spatial extent of change, assessing the relevance of historical experience, and determining which diagnostic evidence is necessary for the adaptation choices under consideration. However, incorporating all available diagnostics can increase the information burden and obscure decision-relevant signals, while compact summaries may omit evidence needed to distinguish among plausible adaptation choices. The first challenge is therefore to form an appropriate adaptation plan with sufficient diagnostic support but without unnecessary information burden.

How to improve adaptation decisions from limited execution feedback? As operating conditions evolve, the effectiveness of adaptation decisions may also change, requiring the decision policy itself to improve from accumulated experience. However, online execution reveals only the outcome of the selected plan, while the outcomes of alternative plans under the same state remain unobserved. This makes it difficult to determine which decisions should be preferred from the same predictor state. Obtaining comparable outcomes requires additional execution, while small differences between realized outcomes may reflect randomness rather than meaningful differences in adaptation effectiveness. The second challenge is therefore to derive informative decision supervision from limited and potentially ambiguous execution feedback for improving future adaptation decisions.

We propose **CoEvoST**, a dual evolution framework for continual spatiotemporal forecasting that couples fast predictor adaptation with slower evolution of an LLM adaptation agent. For **evidence-to-action reasoning**, CoEvoST introduces progressive evidence guided planning, where the agent starts from compact diagnostic summaries and performs multi-round evidence retrieval and plan refinement, ensuring sufficient diagnostic support before the finalized plan is executed to update the predictor. For **outcome-to-policy learning**, CoEvoST executes candidate plans from the same predictor checkpoint and evaluates their realized outcomes to construct near-optimal preference sets, allowing multiple similarly effective plans to be retained as valid preferences. These preferences are then accumulated for online field-weighted DPO to progressively improve the agent's decision policy. Overall, our contributions are summarized as follows:

- We propose CoEvoST, a dual evolution framework that enables both the predictor and the adaptation agent to evolve over time, offering a new perspective on continual spatiotemporal forecasting.
- We develop a progressive evidence guided planning method for evidence-to-action reasoning that adaptively acquires diagnostic evidence and refines adaptation plans before predictor updates.
- We develop an experience driven agent evolution method for outcome-to-policy learning through near-optimal preference construction and online field-weighted DPO.
- Extensive experiments against diverse baselines demonstrate the effectiveness of CoEvoST across four spatiotemporal streams, while further analyses validate the roles of progressive adaptation planning and online agent evolution.

2 Related Work

2.1 Continual and Adaptive Spatiotemporal Forecasting

Continual learning studies how models adapt to sequential tasks or evolving data streams over time. Representative approaches include replay, parameter regularization, parameter isolation or expansion, and prototype- or memory-based adaptation (Kirkpatrick et al., 2017; Liu et al., 2018; Rolnick et al., 2019). Continual spatiotemporal forecasting further involves temporally correlated observations under changing distributions, temporal patterns, node sets, and graph structures.

TrafficStream combines replay, subgraph selection, and regularization for expanding traffic networks (Chen et al., 2021); STKEC and PECPM use influence- or pattern-based memory for knowl-

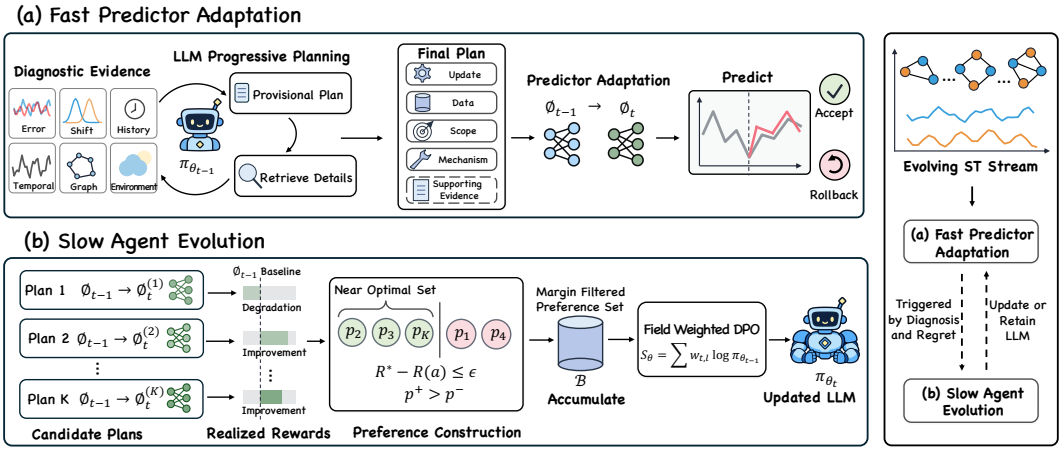


Figure 1: Overview of CoEvoST. The framework couples fast predictor adaptation with slower LLM agent evolution through progressive evidence guided planning and near-optimal preference learning from candidate execution outcomes.

edge consolidation and expansion (Wang et al., 2023a;b); EAC learns a compact continual prompt pool with a frozen backbone (Chen & Liang, 2025); ST-LoRA introduces node-adaptive low-rank updates (Ruan et al., 2025); STRAP retrieves historical, structural, and semantic patterns (Zhang et al., 2025); and ST-TTC performs frequency-domain test-time calibration with streaming memory (Chen & Liang, 2026). These methods mainly develop specialized mechanisms for adapting the predictor, while the higher-level selection of adaptation data, scope, and mechanisms across changing operating states remains less explored.

2.2 LLM Agents and Optimization

LLM agents interact with environments through reasoning, planning, tool use, and feedback. ReAct interleaves reasoning and actions to incorporate environmental observations (Yao et al., 2023), while recent studies have explored agents for data optimization, retrieval, and iterative time-series forecasting (Yeh et al., 2025; Jalori et al., 2025). Agent policies can further be improved from preferences or interaction outcomes, with DPO directly optimizing policies from pairwise preferences (Rafailov et al., 2023) and recent agent-evolution methods exploiting environmental feedback and accumulated experience (Zhao et al., 2024; Wu et al., 2025). In continual forecasting, predictor execution naturally provides adaptation outcomes; CoEvoST uses these outcomes to construct comparable plan preferences and improve the adaptation agent over time.

3 Problem Formulation

We consider continual spatiotemporal forecasting as a sequential decision process jointly driven by a forecasting predictor f_ϕ and an adaptation agent. The agent is implemented by an LLM-based planning policy π_θ, parameterized by θ, which maps the current operating state to an adaptation plan. As a chronological spatiotemporal stream $\{\mathcal{D}_t\}_{t=1}^{T}$ arrives, the predictor continuously performs forecasting and adaptation, while the agent determines how the predictor should adapt under the current condition.

At time t, the incoming chunk $\mathcal{D}_t$ contains node observations X_t, graph structure A_t, active nodes V_t, optional covariates C_t, and a subsequently revealed target Y_t. Before Y_t becomes available, the predictor produces

$$\hat{Y}_t = f_{\phi_{t-1}}\left(X_{\le t}, A_t, C_{\le t}, V_t\right), \tag{1}$$

where ϕ_{t-1} denotes the predictor state before the current adaptation. After the target is observed, the system constructs an operating state s_t from the current forecasting behavior, stream characteristics, and historical adaptation information. Based on this state, the agent generates an adaptation plan

$$p_t \sim \pi_{\theta_{t-1}}(\cdot \mid s_t), \tag{2}$$

where p_t specifies whether to update the predictor, which data to use, the adaptation scope, and the adaptation mechanism.

Executing p_t updates the predictor as $\phi_t = \mathcal{U}(\phi_{t-1}, p_t, \mathcal{D}_{\leq t})$, yielding an adapted predictor for subsequent forecasting. The effectiveness of the executed plan is then evaluated on future observations. Over a longer timescale, accumulated adaptation outcomes are used to optimize the planning policy, $\theta_t = \mathcal{O}(\theta_{t-1}, \mathcal{B}_{\leq t})$, where $\mathcal{B}_{\leq t}$ collects historical operating states, adaptation plans, and their realized outcomes. When no agent update is triggered, $\theta_t = \theta_{t-1}$.

The predictor and the agent therefore evolve at different timescales: the predictor adapts to the changing data stream, while the agent improves its planning policy from accumulated adaptation experience. Continual spatiotemporal forecasting is thus formulated as a dual-evolution process.

4 CoEvoST

We introduce CoEvoST, a dual evolution framework for continual spatiotemporal forecasting that couples a forecasting model with an LLM-based adaptation agent. As illustrated in Fig. 1, the agent performs evidence-to-action reasoning to guide fast predictor adaptation, while execution outcomes are converted into near-optimal preferences for slower agent evolution. Together, the predictor and agent evolve jointly over the data stream.

4.1 Fast Predictor Adaptation: Evidence Driven Continual Adaptation

As data and environments change over time, a previously trained predictor may gradually degrade. In practice, data experts typically examine recent observations from multiple perspectives before deciding how the model should adapt. CoEvoST follows this process. Specifically, it constructs diagnostic evidence from multiple aspects, defines a feasible space of adaptation strategies, and progressively plans the adaptation based on the available evidence.

4.1.1 Diagnostic Evidence and Adaptation Space

CoEvoST represents the current operating state using multi-source diagnostic evidence derived from recent observations, forecasting behavior, structural changes, external conditions, and historical experience. We organize this evidence into six views: prediction state, distribution shift, temporal change, structural change, environmental change, and historical support. In addition, CoEvoST retrieves completed adaptation experiences from historically similar operating states to provide contextual support for the current decision. Based on these signals, the system constructs a compact state summary from diagnostic statuses, risk scores, and key statistics, while retaining the underlying fine-grained evidence for subsequent retrieval. Detailed statistics, diagnostic construction, and experience retrieval are provided in Appendix A.2.

Given the diagnostic state s_t, CoEvoST defines an adaptation strategy as

$$a_t = (u_t, d_t, g_t, m_t), \tag{3}$$

where u_t determines whether to update the predictor, d_t specifies the adaptation data, g_t defines the adaptation scope, and m_t selects an adaptation mechanism. The resulting strategy space contains combinations of data selection, global or local adaptation scope, and representative continual adaptation mechanisms.

Importantly, candidate strategies require different diagnostic evidence for decision support. For example, localized adaptation depends on fine-grained node-level change evidence, whereas retrieval-based adaptation depends on historical similarity and support. CoEvoST therefore associates each strategy with evidence requirements and compatibility constraints, defining a feasible strategy space $\mathcal{A}(s_t)$. Details of the strategy space and constraints are provided in Appendix A.3.

4.1.2 Progressive Evidence Guided Planning

Given the diagnostic state s_t and the strategy space $\mathcal{A}(s_t)$, the remaining challenge is to translate the current state into an executable adaptation plan. This process involves more than selecting a mechanism from fixed statistics. The planner jointly interprets interacting diagnostic factors, requests additional evidence when needed, incorporates relevant historical experience, and coordinates the update decision, data choice, adaptation scope, and mechanism. The resulting adaptation plan p_t

consists of the selected adaptation strategy a_t and its supporting diagnostic evidence $\mathcal{E}_t$. To perform this evidence-aware planning, CoEvoST employs an LLM as the planning core.

However, the complete diagnostic state can contain a large amount of fine-grained evidence, particularly node-level diagnostics and historical information. Providing all such details at once can introduce considerable context overhead and distract the planner with irrelevant information, while relying only on compact summaries may omit evidence needed for fine-grained decisions. To balance these two extremes, CoEvoST adopts a progressive evidence guided planning process: the agent first identifies key diagnostic issues from compact summaries and then selectively retrieves relevant details for further reasoning and plan refinement.

Specifically, the planning process starts from an initial diagnostic summary $c_t^{(0)}$ based on s_t, which provides a compact characterization of the current state based on the available diagnostic evidence. Based on this summary, the agent produces a provisional plan $p_t^{(0)} \sim \pi_{\theta_{t-1}}(\cdot \mid c_t^{(0)})$. CoEvoST then progressively refines this plan through multiple rounds of evidence retrieval and reassessment.

At planning round k, the LLM planner issues an evidence request $\mathcal{R}_t^{(k)}$, which the agent harness supplements when necessary to cover the diagnostic evidence required by the strategy $a_t^{(k)}$. Let $\mathcal{F}_t^{(k)}$ denote the evidence already retrieved. The diagnostic summary is progressively augmented as

$$c_t^{(k+1)} = \mathrm{Augment}\Big(c_t^{(k)}, \mathrm{Retrieve}\Big(\mathcal{R}_t^{(k)} \setminus \mathcal{F}_t^{(k)}\Big)\Big), \qquad p_t^{(k+1)} \sim \pi_{\theta_{t-1}}(\cdot \mid c_t^{(k+1)}), \tag{4}$$

where $\mathcal{R}_t^{(k)}$ denotes the evidence request proposed by the planner and supplemented by the harness when necessary, and $\mathcal{R}_t^{(k)} \setminus \mathcal{F}_t^{(k)}$ denotes the requested evidence that has not yet been retrieved. The updated evidence may support the provisional plan or lead the agent to revise its adaptation strategy and retrieve the evidence required by the new strategy. This process continues until sufficient required diagnostic details have been retrieved and the evidence conditions of the selected adaptation strategy are satisfied, after which the agent finalizes the plan at round K:

$$p_t^{(K)} = \big(a_t^{(K)}, \mathcal{E}_t^{(K)}\big), \tag{5}$$

where $a_t^{(K)}$ is the final adaptation strategy and $\mathcal{E}_t^{(K)}$ records the evidence supporting this decision.

The final plan is then executed to adapt the predictor $f_{\phi_{t-1}}$, yielding a candidate predictor f_{ϕ_t} for the next chunk. Once that chunk's target is observed, validation either retains the candidate or restores the pre-update predictor state $f_{\phi_{t-1}}$ for later forecasts. As the data stream progresses, CoEvoST continues to diagnose the current state and adapt the predictor when needed.

4.2 Slow Agent Evolution: Experience Driven Planning Optimization

Fast predictor adaptation follows the strategy selected by the agent, making future forecasting performance dependent on the quality of adaptation decisions. As the stream evolves, the relationship between diagnostic states and effective adaptation strategies may also change, requiring the planning policy itself to improve over time. CoEvoST therefore accumulates adaptation experience from realized execution outcomes and uses it to continually optimize the agent's planning policy.

4.2.1 Offline Initialization of Planning Capability

To establish the initial planning capability, we construct high-quality demonstrations from historical data and initialize the agent π_{θ_0} through LoRA-based supervised fine-tuning (SFT) (Ouyang et al., 2022). The demonstrations are selected from executed adaptation plans with favorable realized outcomes. They cover both initial planning from diagnostic summaries and subsequent reassessment with retrieved evidence, allowing the agent to learn structured planning and evidence-guided strategy selection. This initialization provides the starting policy for subsequent online agent evolution.

4.2.2 Candidate Exploration and Near-Optimal Preference Construction

Online agent evolution requires converting realized adaptation outcomes into preference supervision. However, online execution only reveals the outcome of the selected adaptation plan, leaving the effectiveness of alternative strategies under the same state unknown. Moreover, we observe that

repeated sampling from the agent given the same state s_t may produce similar adaptation plans, limiting the diversity of alternatives available for comparison. CoEvoST therefore explicitly constructs candidate adaptation plans and compares their realized outcomes to accumulate preference data for online optimization.

Since executing multiple adaptation plans at every state would introduce substantial computational cost and provide limited additional information when the predictor remains stable, CoEvoST selects a sparse set of exploration states

$$\mathcal{T}_{\text{explore}} = \{t \mid \text{Explore}(s_t) = 1\}, \tag{6}$$

where $\text{Explore}(s_t)$ is triggered by substantial prediction degradation, distributional, structural, or environmental changes indicated by the diagnostic state s_t, as well as periodic exploration. To obtain comparable execution outcomes, for each $t \in \mathcal{T}_{\text{explore}}$, each candidate strategy $a_t \in \mathcal{A}(s_t)$ is executed from the same predictor checkpoint using only data available up to time t, and evaluated on the same subsequent unseen window. Candidate branches are restored after evaluation, and only the active agent's selected plan can update the deployed predictor. The realized reward $R_t(a)$ is defined as the relative MAE improvement over the no-update predictor.

We observe that multiple candidate strategies may achieve similarly high realized rewards, with small differences arising from randomness in adaptation and evaluation. Therefore, treating the candidate with the highest reward as the unique preferred strategy may introduce misleading preference supervision. CoEvoST instead identifies a set of near-optimal strategies:

$$R_t^* = \max_{a \in \mathcal{A}(s_t)} R_t(a), \qquad \mathcal{A}_t^+ = \{a \in \mathcal{A}(s_t) \mid R_t^* - R_t(a) \le \epsilon\}, \tag{7}$$

where ϵ controls the tolerance for strategies whose realized rewards are close to the best observed reward. Accordingly, the preference set for state s_t is

$$\mathcal{B}_t = \left\{(s_t, a^+, a^-, R_t(a^+), R_t(a^-)) \;\middle|\; a^+ \in \mathcal{A}_t^+,\ a^- \in \mathcal{A}(s_t) \setminus \mathcal{A}_t^+\right\}, \tag{8}$$

where $a^- \in \mathcal{A}(s_t) \setminus \mathcal{A}_t^+$ denotes a strategy outside the near-optimal set. For preference construction, each candidate strategy is instantiated into a controlled adaptation plan, while the diagnostic summary in s_t and the strategy-specific evidence of the two candidates in each preference pair are combined into a context $\bar{c}_t$ for constructing the preferred and rejected plans p_t^+ and p_t^-.

4.2.3 Online Field-Weighted Preference Optimization

As preference experiences accumulate, CoEvoST triggers online agent optimization when the current decision exhibits sufficiently large empirical regret, $\rho_t = R_t^* - R_t(a_t^{\text{cur}})$, where a_t^{cur} is the strategy selected by the current agent. The triggering conditions are provided in Appendix A.5

The adaptation plan p_t contains multiple structured fields, many of which are shared across competing plans. Treating all output tokens equally may dilute the preference signal from the fields that directly determine the adaptation strategy. We therefore assign different weights to different fields and define the field weighted sequence score as

$$S_\theta(\bar{c}_t, p_t) = \sum_{\ell=1}^{|p_t|} w_{t,\ell} \log \pi_\theta(p_{t,\ell} \mid \bar{c}_t, p_{t,<\ell}), \tag{9}$$

where $p_{t,\ell}$ is the ℓ-th token of the serialized final plan and $w_{t,\ell}$ is its field specific weight. Tokens corresponding to execution-critical the adaptation strategy a_t receive the highest weights, followed by supporting-evidence fields $\mathcal{E}_t$, while structural tokens retain the base weight.

DPO (Rafailov et al., 2023) is well suited to learning the relative effectiveness of adaptation strategies under the same state. Online optimization therefore applies DPO using the field weighted sequence score:

$$\mathcal{L}(\theta) = -\mathbb{E}\left[\log \sigma\left(\beta\left[S_\theta(\bar{c}_t, p_t^+) - S_{\text{ref}}(\bar{c}_t, p_t^+) - S_\theta(\bar{c}_t, p_t^-) + S_{\text{ref}}(\bar{c}_t, p_t^-)\right]\right)\right], \tag{10}$$

where S_{ref} denotes the field weighted sequence score of the pre-update agent, which remains fixed throughout the current DPO optimization.

The optimized agent is treated as a candidate and adopted only if its decisions do not degrade performance on recent states. Otherwise, the previous agent is retained. Adopted agent updates affect only subsequent decisions, while their cumulative forecasting effect is evaluated over the complete chronological stream.

Table 1: Comparison with STD-PLM control strategies. Best results are shown in **bold**.

Method	ENERGY			PEMS			AIR			CAMELS-13		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
Static	1.55	2.39	20.70	22.27	37.06	48.41	17.56	31.46	30.22	14.11	44.35	39.42
Periodic	1.43	1.90	10.46	13.98	22.94	30.50	17.79	31.29	30.89	16.74	50.33	31.88
Triggered	1.11	1.58	8.49	17.43	28.59	36.72	17.90	31.17	31.62	21.48	50.71	32.89
CoEvoST	**1.04**	**1.55**	**7.53**	**13.69**	**22.77**	**25.93**	**17.43**	**30.49**	**30.05**	**13.61**	**44.20**	**28.30**

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Datasets and stream construction. We evaluate CoEvoST on four real-world spatiotemporal streams: ENERGY (Chen & Liang, 2025), PEMS (Chen et al., 2021), AIR (Chen & Liang, 2025), and CAMELS-13 (Addor et al., 2017), covering energy, traffic, air quality, and hydrology, respectively. An initial prefix of each stream is used for offline predictor training, diagnostic calibration, and agent SFT, while the remaining data are evaluated chronologically online. Detailed dataset statistics and stream construction are provided in Appendix A.1.

Models and training. Our primary configuration uses STD-PLM (Huang et al., 2025) as the predictor and Qwen3-8B (Yang et al., 2025) as the agent backbone. We additionally evaluate alternative predictor and LLM backbones in Appendix B.3. The agent is initialized by LoRA-based SFT and subsequently updated through online DPO.

Baselines and evaluation. We compare against static forecasting, periodic and MAE-triggered updates, and representative continual forecasting methods, including TrafficStream (Chen et al., 2021), STKEC (Wang et al., 2023a), PECPM (Wang et al., 2023b), EAC (Chen & Liang, 2025), ST-LoRA (Ruan et al., 2025), STRAP (Zhang et al., 2025), and ST-TTC (Chen & Liang, 2026). We report MAE, RMSE, and MAPE averaged over three random seeds. All online experiments strictly follow chronological data availability: only information observed at the current decision time can be used, and model updates affect only future predictions. Details are in Appendix A.1.

5.2 OVERALL FORECASTING PERFORMANCE

Table 1 shows that keeping the predictor static results in clear performance degradation across all four streams, particularly on PEMS and CAMELS-13, highlighting the need for continual adaptation in evolving environments. Periodic and rule-triggered adaptation substantially improve over the static predictor on ENERGY and PEMS, but provide inconsistent gains on AIR and CAMELS-13, with several metrics even deteriorating after adaptation. This indicates that fixed update rules may be insufficient when adaptation needs vary across operating states. In contrast, CoEvoST achieves more consistent improvements by jointly deciding whether to update and how to adapt according to the diagnosed state.

Table 2 further shows that CoEvoST achieves consistently strong performance against existing continual forecasting baselines across different application domains. Its advantages are particularly evident on ENERGY, PEMS, and CAMELS-13, while it remains competitive on the relatively stable AIR stream. The second-best results are distributed across different baselines and datasets, indicating that individual adaptation mechanisms may perform differently across domains and change patterns. CoEvoST maintains strong performance across the four streams, demonstrating consistent effectiveness under diverse evolving conditions.

5.3 ADAPTATION NEEDS ACROSS STATES AND STAGES

To examine how adaptation needs vary over the stream, we conduct a controlled analysis by executing each candidate adaptation strategy on every online chunk and measuring its forecasting gain. We then rank the strategies within each stage according to their average gains, as shown in Figure 2. The best-performing strategy changes across stages on ENERGY, PEMS, and CAMELS-13, showing that a fixed adaptation strategy cannot consistently match evolving operating conditions. This result supports the motivation for state-dependent adaptation planning.

Table 2: Comparison with continual spatiotemporal forecasting methods. Best results are shown in **bold**, and second-best results are underlined.

Dataset	Metric	TStream	STKEC	PECPM	EAC	ST-LoRA	STRAP	ST-TTC	CoEvoST
ENERGY	MAE	2.50	1.19	1.48	2.71	1.39	2.52	1.52	**1.04**
	RMSE	4.13	1.79	2.26	5.40	2.32	4.84	2.47	**1.55**
	MAPE	28.29	8.42	12.72	23.96	21.71	26.60	13.14	**7.53**
PEMS	MAE	16.36	16.56	15.93	16.82	17.31	18.87	16.30	**13.69**
	RMSE	27.41	28.33	26.59	28.12	29.05	28.60	27.28	**22.77**
	MAPE	28.28	26.11	**25.26**	36.31	27.94	29.44	28.08	25.93
AIR	MAE	17.60	17.85	17.66	17.95	18.16	18.45	17.73	**17.43**
	RMSE	31.13	30.84	30.56	30.55	31.06	31.19	30.54	**30.49**
	MAPE	34.79	35.37	35.21	36.84	33.98	40.65	35.60	**30.05**
CAMELS-13	MAE	28.10	18.58	20.13	17.65	15.76	20.06	24.19	**13.61**
	RMSE	59.90	56.26	59.63	51.48	48.02	58.47	56.73	**44.20**
	MAPE	45.37	37.84	40.85	40.30	35.95	46.64	40.55	**28.30**

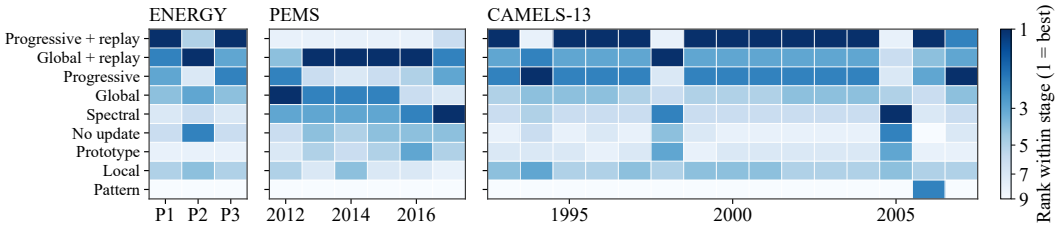


Figure 2: Candidate rankings by mean reward within each stage.

5.4 ABLATION STUDIES

Table 3 evaluates CoEvoST from two complementary perspectives. The first group compares different agent learning paradigms. The base Qwen3-8B performs substantially worse, indicating that pretrained LLM knowledge alone is insufficient for effective state-dependent adaptation planning. Offline SFT provides a strong initialization and leads to clear improvements, while further online DPO yields additional gains, suggesting that the adaptation policy benefits from continued optimization as new operating conditions and execution outcomes accumulate. Replacing the LLM planner with a neural policy network also results in inferior performance, highlighting the value of multi-round evidence retrieval, plan refinement, and historical experience in adaptation planning.

The second group examines the main design components of CoEvoST. Removing progressive evidence retrieval degrades performance, supporting the value of selectively acquiring diagnostic evidence and refining the adaptation plan. Using a single-best preference for each state also performs worse, as strategies with rewards close to the best candidate may be incorrectly treated as rejected examples, introducing noisy preference supervision. Finally, equal-weight DPO weakens optimization because adaptation-related decision fields are diluted by shared structural tokens in the serialized plan, supporting the proposed field-weighted preference objective. Further analysis is provided in Appendix B.2.

5.5 ANALYSIS OF PLANNING AND AGENT EVOLUTION

Evidence-to-action planning. Figure 3 follows a recorded ENERGY decision and illustrates the progressive planning process. The agent refines its provisional plan with additional diagnostic evidence before executing the finalized strategy to update the predictor, illustrating how evidence retrieval supports adaptation planning in practice.

Outcome-to-policy evolution. Figure 4 compares agent decisions before and after online DPO on the same recorded states. After optimization, the agent changes its adaptation strategy on a subset of states, with the paired outcomes showing an overall shift toward lower forecasting error and higher reward. This supports that online DPO improves adaptation decisions on observed states and helps refine the agent's decision policy from accumulated experience.

Table 3: Component and agent comparisons on ENERGY, PEMS, and CAMELS-13.

Method	ENERGY			PEMS			CAMELS-13		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
Base Qwen3-8B	1.35	1.86	10.28	16.27	29.95	31.67	16.93	48.73	32.05
SFT only	1.21	1.63	8.70	14.52	24.34	28.08	14.38	45.97	30.11
Neural policy network	1.23	1.61	8.71	15.03	26.52	29.75	14.72	46.36	34.79
Summary-only planning	1.42	1.97	11.08	16.58	31.17	32.86	15.25	49.10	33.26
Single-best preference	1.36	1.79	10.21	15.82	25.35	30.12	14.82	46.04	33.87
Equal-weight DPO	1.28	1.75	9.03	14.96	25.43	28.81	14.88	46.87	32.45
CoEvoST	**1.04**	**1.55**	**7.53**	**13.69**	**22.77**	**25.93**	**13.61**	**44.20**	**28.30**

1. Initial evidence The latest observed MAE is 2.811, above the rolling MAE of 1.331 and baseline MAE of 0.826. Thirty-one new nodes are present. The agent retrieves three similar, completed experiences in which Progressive + replay improved the next chunk.

2. Provisional plan The agent identifies severe degradation, data and graph shifts, and strong historical support. From this compact evidence, it proposes a light global Progressive + replay update.

3. Required evidence retrieval The proposed strategy requires the error trend and node shift scores. Harness retrieves these fields before the plan can be finalized. The trend is increasing, and 64 of 103 node scores are returned. Nodes 31, 51, 18, 68, and 44 each have a shift score of approximately 0.6931.

4. Reassessment and execution After reviewing the additional evidence, the agent confirms its original plan. Harness validates the plan, and the predictor updates 6,144 trainable parameters.

5. Next chunk validation The updated candidate and a no update shadow are evaluated on the same next chunk. After its label arrives, their MAEs are 1.852 and 2.169, respectively. The historical retention check also passes, so the update is accepted.

Figure 3: Observed progressive planning and validation in Case A from Appendix C.1. Only key information is retained, and the original structured outputs are reformulated for clearer presentation.

Predictor update validation. Across the four streams, 22.6%–43.1% of attempted predictor updates are rolled back after next-chunk validation. For example, 144 of 635 attempted updates on PEMS are rolled back after validation. These results show that online adaptation decisions are not uniformly beneficial, highlighting the importance of execution validation and rollback for autonomous continual forecasting. Detailed statistics are reported in Appendix C.2.

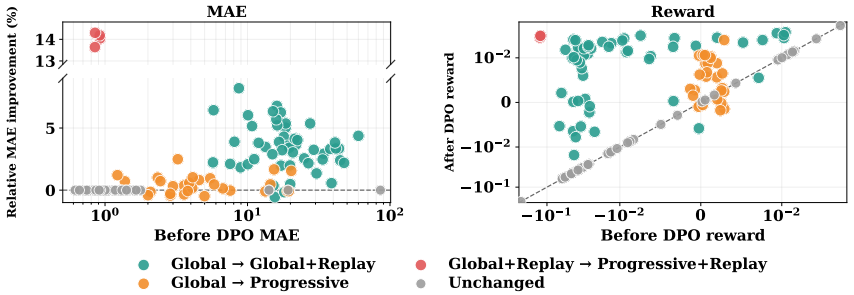


Figure 4: Paired agent decisions before and after online DPO. Left: relative MAE change. Right: reward change. Colors indicate strategy changes; gray denotes unchanged strategies.

6 Conclusion

We presented CoEvoST, a dual-evolution framework for continual spatiotemporal forecasting in which both the predictor and the adaptation agent evolve over time, offering a new perspective on continual forecasting. CoEvoST combines progressive evidence-guided planning for evidence-to-action reasoning with experience-driven agent evolution for outcome-to-policy learning. Experiments on four spatiotemporal streams demonstrate the effectiveness of CoEvoST across diverse evolving environments.

AI Assistance Statement

The main research ideas, motivations, methodology, and experimental design were developed by the authors. Generative AI tools were used only to assist with refining discussions, improving writing, and providing limited code-related support. All technical content, experiments, results, and conclusions were independently reviewed and verified by the authors, who take full responsibility for the final manuscript.

Reproducibility Statement

To support reproducibility, we provide the source code, comments, README documentation, necessary baseline implementations, running scripts, hyperparameter configurations, and data preprocessing procedures in the supplementary material. All datasets used in this work are publicly available. Detailed experimental settings, including data splits, training and adaptation configurations, model settings, and implementation details, are provided in the appendix.

References

Nans Addor, Andrew J Newman, Naoki Mizukami, and Martyn P Clark. The camels data set: catchment attributes and meteorology for large-sample studies. *Hydrology and Earth System Sciences*, 21(10):5293–5313, 2017.

Wei Chen and Yuxuan Liang. Expand and compress: Exploring tuning principles for continual spatio-temporal graph forecasting. In *International Conference on Learning Representations*, volume 2025, pp. 81631–81656, 2025.

Wei Chen and Yuxuan Liang. Learning with calibration: Exploring test-time computing of spatio-temporal forecasting. *Advances in Neural Information Processing Systems*, 38:155895–155929, 2026.

Xu Chen, Junshan Wang, and Kunqing Xie. TrafficStream: A streaming traffic flow forecasting framework based on graph neural networks and continual learning. In Zhi-Hua Zhou (ed.), *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pp. 3620–3626. International Joint Conferences on Artificial Intelligence Organization, August 2021. doi: 10.24963/ijcai.2021/498.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Yiheng Huang, Xiaowei Mao, Shengnan Guo, Yubin Chen, Junfeng Shen, Tiankuo Li, Youfang Lin, and Huaiyu Wan. Std-plm: Understanding both spatial and temporal properties of spatial-temporal data with plm. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pp. 11817–11825, 2025.

Gunjan Jalori, Preetika Verma, and Sercan Ö. Arik. FLAIRR-TS - forecasting llm-agents with iterative refinement and retrieval for time series. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025, Suzhou, China, November 4-9, 2025*, pp. 15427–15437. Association for Computational Linguistics, 2025. URL https://doi.org/10.18653/v1/2025.findings-emnlp.834.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Xialei Liu, Marc Masana, Luis Herranz, Joost Van de Weijer, Antonio M. López, and Andrew D. Bagdanov. Rotate your networks: Better weight consolidation and less catastrophic forgetting. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pp. 2262–2268, 2018. doi: 10.1109/ICPR.2018.8545895.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744. Curran Associates, Inc., 2022. doi: 10.52202/068431-2011.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *Advances in neural information processing systems*, 32, 2019.

Weilin Ruan, Wei Chen, Xilin Dang, Jianxiang Zhou, Weichuang Li, Xu Liu, and Yuxuan Liang. St-lora: Low-rank adaptation for spatio-temporal forecasting. In Rita P. Ribeiro, Bernhard Pfahringer, Nathalie Japkowicz, Pedro Larrañaga, Alípio M. Jorge, Carlos Soares, Pedro H. Abreu, and João Gama (eds.), *Machine Learning and Knowledge Discovery in Databases. Research Track - European Conference, ECML PKDD 2025, Porto, Portugal, September 15-19, 2025, Proceedings, Part VII*, volume 16019 of *Lecture Notes in Computer Science*, pp. 345–361. Springer, 2025. URL https://doi.org/10.1007/978-3-032-06109-6_20.

Zezhi Shao, Zhao Zhang, Fei Wang, Wei Wei, and Yongjun Xu. Spatial-temporal identity: A simple yet effective baseline for multivariate time series forecasting. In *Proceedings of the 31st ACM international conference on information & knowledge management*, pp. 4454–4458, 2022.

Binwu Wang, Yudong Zhang, Jiahao Shi, Pengkun Wang, Xu Wang, Lei Bai, and Yang Wang. Knowledge expansion and consolidation for continual traffic prediction with expanding graphs. *IEEE Transactions on Intelligent Transportation Systems*, 24(7):7190–7201, 2023a.

Binwu Wang, Yudong Zhang, Xu Wang, Pengkun Wang, Zhengyang Zhou, Lei Bai, and Yang Wang. Pattern expansion and consolidation on evolving graphs for continual traffic prediction. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2223–2232, 2023b.

Xiaoyang Wang, Yao Ma, Yiqi Wang, Wei Jin, Xin Wang, Jiliang Tang, Caiyan Jia, and Jian Yu. Traffic flow prediction via spatial temporal graph neural network. In *Proceedings of the web conference 2020*, pp. 1082–1092, 2020.

Rong Wu, Xiaoman Wang, Jianbiao Mei, Pinlong Cai, Daocheng Fu, Cheng Yang, Licheng Wen, Xuemeng Yang, Yufan Shen, Yuxin Wang, et al. Evolver: Self-evolving llm agents through an experience-driven lifecycle. *arXiv preprint arXiv:2510.16079*, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Chin-Chia Michael Yeh, Vivian Lai, Uday Singh Saini, Xiran Fan, Yujie Fan, Junpeng Wang, Xin Dai, and Yan Zheng. Empowering time series forecasting with llm-agents. In *2025 IEEE International Conference on Big Data (BigData)*, pp. 2685–2688. IEEE, 2025.

Haoyu Zhang, Wentao Zhang, Hao Miao, Xinke Jiang, Yuchen Fang, and Yifan Zhang. STRAP: Spatio-temporal pattern retrieval for out-of-distribution generalization. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=Y5wMoIbdDs.

Qi Zhao, Haotian Fu, Chen Sun, and George Konidaris. Epo: Hierarchical llm agents with environment preference optimization. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 6401–6415, 2024.

APPENDIX

A EXPERIMENTAL PROTOCOL AND IMPLEMENTATION

A.1 EXPERIMENTAL SETUP

Datasets and stream construction. We evaluate CoEvoST on four real-world spatiotemporal streams covering wind energy (ENERGY (Chen & Liang, 2025)), traffic (PEMS (Chen et al., 2021)), air quality (AIR (Chen & Liang, 2025)), and hydrology (CAMELS-13 (Addor et al., 2017)), containing 4, 7, 4, and 35 periods, respectively. These streams collectively exhibit distributional drift, periodic variation, and structural changes. CAMELS-13 additionally provides meteorological covariates, which are used only for environmental diagnosis and are not provided as inputs to the forecasting predictor. Detailed dataset statistics and stream construction are provided in Table 4.

Table 4: Streaming datasets.

Dataset	Domain	Periods	Nodes	Time Span
ENERGY	Wind energy	4	$103 \rightarrow 134$	240 days
PEMS	Traffic	7	$655 \rightarrow 871$	2011–2017 Jul.–Aug.
AIR	Air quality	4	$1087 \rightarrow 1202$	2016–2019
CAMELS-13	Hydrology(+weather)	35	7	1980–2014

For ENERGY, PEMS, and AIR, the first period is used as the initial offline stage, corresponding to 60 days, 53 days, and one year, respectively. For CAMELS-13, the first 10 periods (10 years) are used for offline initialization. The remaining 180 days, 316 days, 3 years, and 25 years constitute the online streams, respectively. Within the offline stage, data are chronologically divided with a ratio of 4:1:5 for predictor training, predictor validation and diagnostic-threshold calibration, and collection of trajectories for agent SFT, respectively. Normalization statistics are fitted only on the initial predictor-training prefix.

Predictors, LLM, and implementation. Our primary implementation uses STD-PLM (Huang et al., 2025) as the forecasting predictor and Qwen3-8B (Yang et al., 2025) as the planning backbone. To examine backbone generality, we additionally evaluate STGNN (Wang et al., 2020) and STID (Shao et al., 2022) as alternative predictors, and Meta-Llama-3.1-8B-Instruct (Grattafiori et al., 2024) as an alternative agent backbone, with results reported in Appendix B.3.

The initial agent is trained by LoRA-based SFT using trajectories collected during the offline stage, followed by online DPO for agent evolution. Experience retrieval returns the top three completed experiences with similarity scores of at least 0.8. Online DPO becomes eligible after accumulating 24 valid candidate groups and is performed when the conditions in Section 4.2.3 are satisfied.

Both initial predictor training and subsequent predictor adaptation use AdamW with a learning rate of 1×10^{-3} and early-stopping patience of 10. Offline LoRA SFT uses a learning rate of 2×10^{-4}, while online DPO uses 1.5×10^{-5}, both with AdamW; the DPO coefficient β is set to 1. The historical input and forecasting horizons are both 12 steps. Experiments are conducted on an NVIDIA A40 GPU with CUDA 13.0 and PyTorch 2.12.1. Additional training configurations, adaptation budgets, and implementation details are provided in the appendix.

All online experiments strictly follow chronological data availability: a prediction can use only information available at its decision time, adaptation can use only subsequently observed labels, and an updated model is applied only to future predictions. This protocol prevents future information from leaking into either adaptation or evaluation.

Baselines and evaluation metrics. We compare CoEvoST with static forecasting, periodic updates, MAE-triggered updates, and existing continual or adaptive forecasting methods. Static forecasting keeps the initially trained predictor fixed throughout the online stream. Periodic updates apply a fixed adaptation mechanism every 12 hours on ENERGY and PEMS, every day on AIR, and every 180 days on CAMELS-13. Rule-triggered updates apply the same fixed mechanism when recent forecasting performance exhibits sufficient degradation according to the MAE-based trigger.

Continual or adaptive baselines include TrafficStream (Chen et al., 2021) (TStream in the tables), STKEC (Wang et al., 2023a), PECPM (Wang et al., 2023b), EAC (Chen & Liang, 2025), ST-LoRA (Ruan et al., 2025), STRAP (Zhang et al., 2025), and ST-TTC (Chen & Liang, 2026). Each method retains its central adaptation mechanism and continual update rule. Key hyperparameters are tuned around the recommended or best-performing settings reported in the corresponding papers. We report mean absolute error (MAE), root mean squared error (RMSE), and mean absolute percentage error (MAPE), where MAE and RMSE are measured in the original data units and MAPE is reported as a percentage. All results are averaged over three random seeds.

A.2 Detailed Diagnostic Evidence

CoEvoST characterizes each operating state using raw observation statistics and six views of high-level diagnostic evidence: prediction state, distribution shift, historical support, temporal change, structural change, and environmental change. Table 6 summarizes the detailed evidence used for each view. For each decision point, the complete metric-level diagnostic outputs are retained, while a compact summary is constructed using fixed field-selection rules. Each diagnostic view is represented by a diagnostic status, a risk score, a short description, and a small set of representative statistics. Diagnostic levels are calibrated using q_{50}, q_{75}, and q_{90} estimated only from the offline data. High-dimensional node-level evidence is summarized using the five nodes with the largest changes, while raw observation distributions are represented by the 5th, 25th, 50th, 75th, and 95th percentiles together with trend and volatility statistics. The complete diagnostic outputs remain linked to their summaries so that fine-grained evidence can be retrieved during subsequent planning. Missing or failed diagnostics are explicitly marked as unavailable rather than imputed.

Historical support and historical experience retrieval serve different purposes. Historical support is a diagnostic signal that characterizes the relevance and consistency of past observations for the current state, whereas experience retrieval returns completed adaptation records from similar historical states. A decision is added to the experience pool only after its execution outcome becomes available. For retrieval, the current state and each historical state are represented by a six-dimensional risk profile corresponding to the six diagnostic views, and historical experiences are ranked by cosine similarity. The current implementation retrieves at most three experiences with similarity no lower than 0.8, including successful, failed, and rolled-back adaptations when applicable. Each retrieved experience contains a compact record of the historical diagnostic assessment, selected adaptation strategy, execution outcome, and realized gain rather than the original historical tensors.

Table 5: Overview of high-level diagnostic evidence in CoEvoST.

Diagnostic view	**Representative evidence**
Prediction state	Recent and historical errors, error trends, and node-level error
Distribution shift	Distribution discrepancy and node-level distribution changes
Historical support	Historical similarity, support strength, and outcome consistency
Temporal change	Periodic, spectral, and sampling-pattern changes
Structural change	Node, edge, and graph-topology changes
Environmental change	Changes in external covariates and extreme conditions

Table 6: Detailed diagnostic evidence used to characterize the operating state in CoEvoST. Environmental evidence is included only when the corresponding covariates are available.

Diagnostic view	Representative evidence	Diagnostic information
Prediction state	Rolling MAE, RMSE, and MAPE; error trends; signed bias; node-level errors and error concentration	Characterizes recent and historical predictive performance, performance degradation, and whether errors are concentrated on particular nodes
Distribution shift	RBF-MMD; representation cosine distance; node-level Jensen–Shannon divergence	Characterizes the magnitude and spatial distribution of changes between recent and reference data distributions
Historical support	Neighbor similarity; effective support count; dispersion and directional agreement of observed historical subsequent values	Characterizes the availability, strength, and consistency of relevant historical experience for the current state
Temporal change	Sampling intervals and gaps; periodic-profile differences; spectral phase and dominant-frequency changes	Characterizes changes in temporal organization, including periodicity, frequency patterns, and sampling behavior
Structural change	Added or removed nodes; shared-edge changes; degree distributions; connected components; isolated nodes	Characterizes changes in graph structure and their spatial extent across the forecasting system
Environmental change	Covariate MMD; standardized mean shifts; extreme-value fractions; missingness changes	Characterizes changes and anomalies in available environmental conditions associated with the current observations

A.3 Candidate Strategies and Evidence Contracts

Given the diagnostic state, CoEvoST selects an adaptation strategy that specifies whether and how the predictor should adapt. We define the core adaptation strategy as

$$a_t = (u_t, d_t, g_t, m_t), \tag{11}$$

where u_t, d_t, g_t, and m_t denote the update decision, adaptation data, adaptation scope, and adaptation mechanism, respectively. The adaptation mechanism m_t is instantiated with representative continual forecasting mechanisms, including global low-rank adaptation (Global) (Ruan et al., 2025), local node adaptation (Local) (Ruan et al., 2025), spectral calibration (Spectral) (Chen & Liang, 2026), progressive LoRA expansion (Progressive) (Chen & Liang, 2025), prototype-based adaptation (Prototype) (Wang et al., 2023a;b), and retrieval-based pattern fusion (Pattern) (Zhang et al., 2025).

Each candidate strategy is associated with compatibility constraints and an evidence contract specifying the diagnostic information required before execution. For example, local adaptation requires node-level shift evidence, whereas retrieval-based adaptation requires historical-support evidence. These constraints define the feasible strategy space $\mathcal{A}(s_t)$ under compatibility constraints.

Table 7 maps the short labels used in the candidate comparisons to their full execution strategy names. Global + replay and Progressive + replay change the data choice of their respective strategies, not the update mechanism.

The data choice `recent` uses labeled data available at the decision time, `recent_replay` adds recent historical batches, and `retrieved_history` uses retrieved historical batches. Global and Progressive allow `recent` or `recent_replay`. Local and Spectral use `recent`, Prototype requires `recent_replay`, and Pattern requires `retrieved_history`. No update uses no adaptation data. The only effective update scopes are `global` and `changed_nodes`. Local uses `changed_nodes`; the other updating strategies use `global`. No update has no update scope.

The evidence contract specifies the diagnostic families that must be available and the detail fields retrieved before the agent finalizes a plan.

Table 7: Execution strategies, experimental short labels, implementation ideas, and required diagnostic evidence.

Execution strategy	Short label	Main implementation idea	Required diagnosis and detail evidence
`no_update`	No update	Keep the predictor unchanged.	Read `error.mae` and `error.trend` to check the decision.
`global_lora_adaptation`	Global	Incrementally train the global predictor's LoRA parameters.	Requires error diagnosis. Read `error.mae` and `error.trend`.
`local_node_adaptation`	Local	Select changed nodes by node shift scores, add neighbors and a few stable nodes, and train with a node masked loss.	Requires error and shift diagnoses. `shift.node_js_scores` must be available and cited. Read `error.node_mae` and `shift.node_js_scores`.
`spectral_calibration`	Spectral	Train output amplitude and phase calibrators without broadly updating the backbone.	Requires error and temporal diagnoses. Read `error.mae`, `temporal.spectral_phase_shift`, and `temporal.dominant_frequency_change`.
`progressive_lora_expansion`	Progressive	Add a LoRA adapter, freeze earlier adapters, and train only the new adapter.	Requires error and shift diagnoses. Read `error.trend` and `shift.node_js_scores`.
`prototype_memory_adaptation`	Prototype	Learn a bounded set of prototypes and fuse retrieved prototype information into node representations.	Requires error and support diagnoses. Read `error.mae`, `support.top_k_similarity`, and `support.historical_future_normalized_dispersion`.
`pattern_retrieval_fusion`	Pattern	Encode completed historical data in a bounded pattern memory, then retrieve and fuse similar patterns at prediction time without gradient training.	Requires support diagnosis. Read `support.top_k_similarity`, `support.nearest_similarity`, and `support.effective_support_count`.

A.4 Online Exploration and Preference Construction

Table 8: Default settings for exploration and preference construction.

Setting	Default
Degradation exploration threshold	30% MAE increase
Periodic exploration interval	12 chunks
Exploration cooldown	3 chunks
Near-optimal tolerance ϵ	0.005
Maximum preference pairs per state	4

Preference pairs follow the set $\mathcal{B}_t$ defined in Section 4.2.2, with preferred strategies drawn from $\mathcal{A}_t^+$ and rejected strategies drawn from its complement. The default near-optimal tolerance is $\epsilon = 0.005$. Before online DPO, duplicate pairs are removed, and sampling gives greater exposure to pairs with larger utility separation and those correcting the current agent decision, while bounding the contribution of strongly negative cases.

Table 9: Default settings for online agent optimization.

Setting	Default
Minimum candidate groups before optimization	24
Minimum new states with reliable preferences	12
Minimum high-regret states	4
Minimum unique preference pairs	24
DPO β / learning rate	1 / 1.5×10^{-5}
Epochs / gradient accumulation	4 / 4
Execution / diagnosis / other token weights	8 / 2 / 1
Agent adoption	Non-inferior on recent reliable states

A.5 Online DPO and Version Adoption

Table 9 summarizes the default configuration for online agent optimization, including the accumulated experience required to trigger DPO, the field-wise token weighting used during optimization, and the criterion for adopting the updated agent. DPO is performed only when sufficient preference experience has accumulated, and the updated agent is accepted only when it does not degrade performance on recent evaluation states.

A.6 Candidate-Version Checks

After online DPO, the updated agent is evaluated against the active version on recent historical states with matched realized outcomes. The active agent uses its recorded decisions, while the updated agent replans under the corresponding historical contexts. The updated version is adopted only when it does not degrade plan validity, outcome coverage, or forecasting performance; otherwise, the previous agent is retained.

B Supplementary Experimental Results

B.1 Candidate Rewards and Oracle Comparison

Figure 5 summarizes mean reward differences across candidate plans. The chunk oracle selects the best candidate after observing all outcomes. It provides an upper bound on the decision opportunity within the predefined candidate set rather than a deployable policy. Table 10 summarizes this gap against the best fixed plan. The best fixed plan differs across datasets. The oracle gap is largest on CAMELS-13. AIR has a smaller gap and weak preference separation, with no update frequently near-optimal. These results complement the stage rankings in Figure 2.

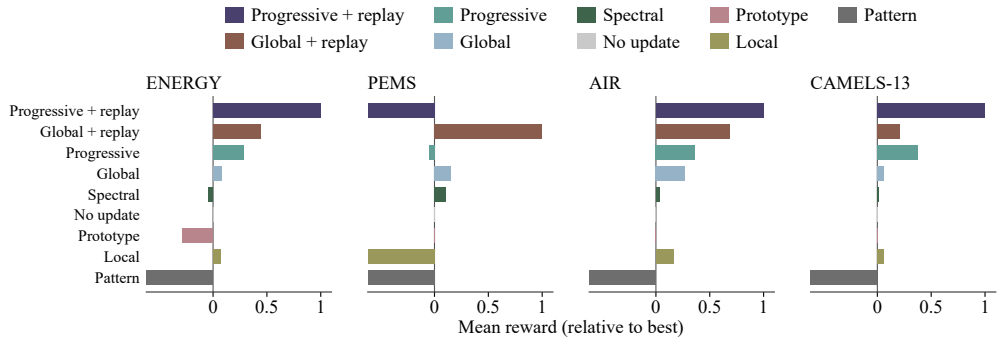


Figure 5: Mean candidate reward relative to the best candidate within each dataset. Candidate comparisons use the same predictor checkpoint within each state. Reward is the next chunk MAE improvement relative to no update.

Controlled candidate execution summary. The summary below uses states where all nine candidates were executable. Fixed and oracle values are averages of the candidate reward used by this analysis. The 0.005 near-optimal threshold is used only for the descriptive candidate analysis; online preference construction uses the threshold specified in Section A.4.

Table 10: Controlled candidate execution across online states. Fixed and oracle rewards are averaged over states where all candidate strategies are available.

Dataset	Best fixed plan	States	Fixed	Oracle	Gap	No-update (%)	Near-opt.
ENERGY	Progressive + replay	358	0.0265	0.0376	0.0111	40.5	3.77
PEMS	Global + replay	634	0.0064	0.0125	0.0060	57.4	4.51
AIR	Progressive + replay	1060	0.0007	0.0022	0.0014	96.6	7.86
CAMELS-13	Progressive + replay	295	0.0204	0.0511	0.0307	39.7	3.90

B.2 Extended Ablation Analysis

Ablation variants. We consider six variants to isolate different aspects of CoEvoST. Base Qwen3-8B uses the pretrained LLM without task-specific agent training. SFT only retains the offline LoRA-based supervised initialization but removes subsequent online DPO. Neural policy network replaces the LLM planner with a learned state-to-action policy. Summary-only planning removes progressive evidence retrieval and makes decisions directly from the initial diagnostic summary. Single-best preference constructs preference supervision using only the highest-reward candidate as preferred, without preserving near-optimal alternatives. Equal-weight DPO replaces the proposed field-weighted objective with uniform token weighting.

To make the relative effect of each design choice more explicit, Table 11 reports the MAE improvement of the full CoEvoST model over each variant.

Agent initialization and online evolution. The comparison among Base Qwen3-8B, SFT only, and CoEvoST separates the effects of task-specific initialization and subsequent online evolution. Base Qwen3-8B performs consistently worse, with CoEvoST reducing MAE by 23.0%, 15.9%, and 19.6% on ENERGY, PEMS, and CAMELS-13, respectively. This suggests that pretrained LLM knowledge alone is insufficient for effective state-dependent adaptation planning in this setting.

Offline SFT substantially narrows this gap, indicating that trajectories collected during initialization provide useful supervision for aligning the agent with the available adaptation strategies. CoEvoST nevertheless further improves over SFT only by 14.0%, 5.7%, and 5.4% in MAE. Since the main difference is the subsequent online preference optimization, these results are consistent with the intended role of DPO in refining the decision policy as new states and realized adaptation outcomes accumulate along the stream. The larger gain on ENERGY further indicates that the benefit of online evolution varies across operating streams.

LLM planning versus a direct policy network. Replacing the LLM planner with a neural policy network leads to higher forecasting error across all three datasets, with MAE gaps of 15.4%, 8.9%, and 7.5%. While both approaches can select adaptation actions from diagnostic states, the LLM-based agent additionally operates over structured diagnostic evidence, retrieved experiences, and multi-round planning contexts. These results support the use of the LLM-based planner within the proposed adaptation framework.

Progressive evidence acquisition. Summary-only planning produces one of the largest degradations among the ablations. Relative to this variant, CoEvoST improves MAE by 26.8% on ENERGY, 17.4% on PEMS, and 10.8% on CAMELS-13. Under summary-only planning, the agent must commit to an adaptation decision using only the compressed initial diagnosis. In contrast, progressive planning can request more detailed evidence when the initial summary is insufficient and revise the plan after receiving that evidence.

The consistent degradation suggests that the compact summary does not always retain enough information to distinguish among competing adaptation strategies. At the same time, progressively retrieving only relevant details avoids exposing the planner to the entire diagnostic state at once.

Table 11: Relative MAE improvement (%) of CoEvoST over different ablation variants.

Variant	ENERGY	PEMS	CAMELS-13
Base Qwen3-8B	23.0	15.9	19.6
SFT only	14.0	5.7	5.4
Neural policy network	15.4	8.9	7.5
Summary-only planning	26.8	17.4	10.8
Single-best preference	23.5	13.5	8.2
Equal-weight DPO	18.8	8.5	8.5

The results are therefore consistent with the motivation for evidence-guided planning: maintaining a compact initial context while preserving access to fine-grained signals when needed.

Preference construction. The Single-best preference variant treats only the highest-reward candidate as preferred. CoEvoST improves over this variant by 23.5%, 13.5%, and 8.2% in MAE across the three datasets. In many online states, however, several candidate strategies produce similar realized rewards. Treating every non-maximum candidate as rejected may therefore impose sharp preference boundaries between actions with nearly equivalent outcomes.

CoEvoST instead preserves near-optimal candidates when constructing preference supervision. The observed improvement supports this tolerance-aware design, which avoids strongly penalizing adaptation decisions whose realized utility is close to the best available outcome.

Field-weighted DPO. Replacing field-weighted optimization with standard equal-weight DPO also consistently degrades performance. CoEvoST yields MAE improvements of 18.8%, 8.5%, and 8.5% over Equal-weight DPO on ENERGY, PEMS, and CAMELS-13, respectively.

This difference is particularly relevant for structured plans. A serialized plan contains both decision-critical fields, such as the selected adaptation strategy and supporting evidence, and many shared structural or formatting tokens. Under uniform token weighting, gradients from these shared tokens can occupy a substantial fraction of the sequence-level objective even though they do not determine the executed adaptation. Field weighting instead emphasizes tokens corresponding to the adaptation decision and diagnostic evidence, concentrating the preference signal on the parts of the completion most relevant to downstream execution. The consistent gap between the two variants supports this structured optimization design.

Overall observations. The ablations show that no single component accounts for the full improvement of CoEvoST. Offline SFT provides the initial task alignment, online DPO further adapts the decision policy to newly observed outcomes, progressive evidence retrieval supports more informed planning, tolerance-aware preference construction reduces supervision from near-equivalent candidates, and field weighting focuses optimization on decision-relevant plan content. Performance decreases when each component is individually removed or replaced across all three datasets, suggesting that these mechanisms provide complementary contributions to the continual agent adaptation process.

B.3 Predictor and Agent Backbones

Table 12 evaluates the extensibility of CoEvoST by replacing one component at a time on ENERGY and PEMS. With Qwen3-8B (Yang et al., 2025) as the agent, we compare STD-PLM (Huang et al., 2025), STGNN (Wang et al., 2020), and STID (Shao et al., 2022) as predictors; with STD-PLM fixed, we compare Qwen3-8B and Meta-Llama-3.1-8B-Instruct (Grattafiori et al., 2024) as agent backbones, following the same evaluation protocol as the main comparison.

Table 12: Predictor and agent backbone replacement on ENERGY and PEMS.

Dataset	Predictor	LLM	MAE	RMSE	MAPE (%)
ENERGY	STD-PLM	Qwen3-8B	1.04	1.55	7.53
ENERGY	STGNN	Qwen3-8B	1.44	1.92	14.78
ENERGY	STID	Qwen3-8B	0.90	1.41	6.09
ENERGY	STD-PLM	Llama 3.1 8B	1.12	1.63	9.98
PEMS	STD-PLM	Qwen3-8B	13.69	22.77	25.93
PEMS	STGNN	Qwen3-8B	18.77	28.81	29.43
PEMS	STID	Qwen3-8B	15.33	25.66	28.96
PEMS	STD-PLM	Llama 3.1 8B	14.58	24.32	27.80

All eight configurations complete online forecasting on the two selected streams. These runs demonstrate that the predictor and agent components can be replaced within the CoEvoST interface. Their absolute errors reflect the chosen backbones as well as the adaptation process.

C Operational Case Records

C.1 Candidate Matching and Recorded Decisions

Table 13 presents four representative online decisions with distinct outcomes. In Case A, the best other candidate was Progressive adaptation without replay, and the selected update was accepted. In Case B, the same alternative improved MAE slightly while the selected update was rolled back. Case C selected a light global update under degraded forecasts and low historical support; the best other candidate was local adaptation. Case D selected no update under stable forecasts. Spectral calibration had the lowest MAE, but its reward advantage was below the near-optimal tolerance used for this case analysis. The AIR examples establish that different actions occur in recorded states, while their small outcome differences do not establish a general advantage from action switching.

In the complete ENERGY operational trace containing Cases A and B, 79 states have outcomes available for all nine candidates. The active plan achieves the highest reward in 32 states and falls within a 0.005 near-optimal band in 45 states. Three online DPO updates are promoted during the trace. These results suggest that the agent's selected plans are frequently competitive with the best available candidates, while the promoted DPO updates indicate that its decision policy continues to adapt as new preference evidence accumulates.

Table 13: Next-chunk MAE of selected plans in four representative online cases. Best other denotes the best alternative candidate, and rank is computed among all available plans.

Case	Dataset	Selected plan	No update	Selected	Best other	Rank
A	ENERGY	Progressive + replay	2.17	1.85	2.07	1
B	ENERGY	Progressive + replay	2.44	2.62	2.42	7
C	AIR	Global LoRA, recent	16.47	16.44	16.46	1
D	AIR	No update	17.0141	17.0141	17.0140	2

C.2 Validation and rollback

Table 14 summarizes predictor update outcomes across the complete online runs. A substantial proportion of attempted updates are accepted across all datasets, while 22.6%–43.1% are rolled back after next-chunk validation. These results indicate that online adaptation is not uniformly beneficial and that some selected updates can degrade subsequent forecasting performance. Validation and rollback therefore provide an important execution safeguard by preventing unsuccessful updates from propagating to later predictor states while retaining beneficial adaptations.

Table 14: Predictor update outcomes in complete valid online runs. Validation and rollback are performed after the next-chunk label arrives to prevent harmful updates.

Dataset	Attempts	Accepted	Rolled back	Rollback rate
ENERGY	357	203	154	43.1%
PEMS	635	491	144	22.6%
AIR	347	232	115	33.1%
CAMELS-13	296	192	104	35.1%

D DISCUSSION

CoEvoST suggests that continual spatiotemporal forecasting can benefit from treating adaptation as a decision problem rather than relying on a fixed update rule or a single adaptation mechanism. The experimental results show that adaptation needs vary across operating states and stages, while the case studies further indicate that diagnostic evidence can refine adaptation plans and that execution experience can be used to improve the agent's decision policy. These observations support the broader view that long-running forecasting systems should adapt not only their predictor parameters, but also the policy that determines when and how adaptation is performed.

The framework also introduces additional computation beyond conventional continual forecasting. Progressive evidence retrieval increases planning overhead, and candidate execution for preference construction requires extra predictor evaluations at selected exploration states. CoEvoST limits these costs through compact diagnostic summaries, selective evidence retrieval, sparse candidate exploration, and less frequent agent updates, but the trade-off between adaptation quality and computational efficiency remains important in large-scale or latency-sensitive deployments.

Several directions remain open. The current strategy space is defined from a fixed library of adaptation mechanisms, and the evidence requirements associated with these strategies are manually specified. Future work could expand the strategy library, learn evidence requirements from accumulated experience, and improve preference construction under delayed or ambiguous outcomes. Extending the framework to richer multimodal environmental signals and broader forecasting backbones may further improve its applicability to continually evolving spatiotemporal systems.