ADAPTIVE SPECTRAL-KOOPMAN DYNAMICS MODELING FOR TEMPORAL DOMAIN GENERALIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Temporal Domain Generalization (TDG) has emerged to address real-world streaming data with distribution shifts over time. However, existing methods are either prone to overfitting to domain-specific noise in the data space or become overly complex and less interpretable in the parameter space. To bridge these gaps, we propose **AdaSpecK**, a spectral-Koopman framework with adaptive context extraction for TDG. To mitigate noise fitting to irregularly sampled domains, we introduce spectral-regularized Koopman dynamics modeling, which applies spectral-aware filtering in the latent space to extract denoised low-frequency trajectories and learn a Koopman operator to model the system dynamics in a linearized space. To model complex historical environments under non-stationarity, we design a context-informed heterogeneous pattern extraction mechanism. Specifically, we employ a target-conditioned attention module to attend to distinct past windows, producing a dynamic, target-specific historical summary. By constructing an environmental signature from the current evolutionary pattern, our model adaptively perceives which aspects of the past context are most informative for future prediction via a learned router. Extensive experiments on eight diverse classification and regression benchmarks demonstrate that AdaSpecK achieves state-of-the-art performance. The code and datasets are available at https://anonymous.4open.science/r/Ada-Spec-K.

1 INTRODUCTION

Conventional machine learning typically assumes that training and test data are independently and identically distributed (IID), which expects a model to generalize across training and test sets. However, this assumption is rarely satisfied in real-world applications where data often arrive as continuous streams and their underlying distributions shift dynamically over time (Han et al., 2024). For instance, in financial markets, trading behaviors evolve as economic conditions change; in social media, linguistic trends and topics shift rapidly over time. This phenomenon, known as concept drift (Lu et al., 2019; Wang et al., 2025), leads models trained on past data to degrade when deployed in distributionally shifted environments (Gama et al., 2014). This has motivated the emergence of Temporal Domain Generalization (TDG), which aims to leverage evolving historical data to build models capable of generalizing to near-future domains (Cai et al., 2024).

Existing approaches to TDG can be broadly categorized into data-centric and model-centric paradigms. Data-centric methods operate on the premise that temporal robustness can be achieved at the data or feature level. They employ strategies such as data augmentation (Liu et al., 2018) or generative modeling to simulate potential future distributions (Zeng et al., 2024), or leverage representation learning (Gong et al., 2019) to extract domain-invariant features and disentangle domain-shared features from domain-specific ones. In contrast, model-centric approaches focus on adapting to evolving domains by dynamically predicting or extrapolating model parameters. These methods utilize techniques such as gradient interpolation (Nasery et al., 2021) or probabilistic frameworks to characterize latent evolutionary dynamics within the parameter space (Cai et al., 2024; Yu et al., 2025). Despite these advances, several critical challenges remain unresolved, as shown in Figure 1:

Challenge 1: Time-irregularly distributed training domains lead to noise fitting when modeling domain characteristics. Unlike conventional forecasting tasks with regularly sampled data, TDG often involves observations at irregular timestamps. This results in training domains that are

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.

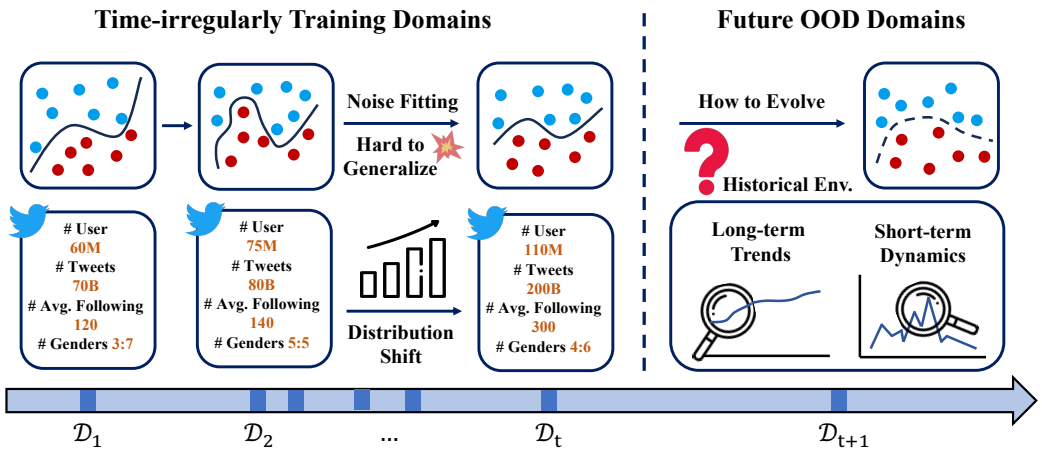


Figure 1: In TDG, training domains are often time-irregularly sampled; directly learning in the data space tends to overfit domain-specific noise, impairing generalization. Predicting future OOD domains requires modeling dynamic temporal patterns across evolving historical environments.

sparsely and unevenly distributed across a continuous timeline. Directly learning a predictive model in the data space risks overfitting to domain-specific patterns or noise, hindering reliable generalization to future out-of-distribution (OOD) domains (Lu et al., 2019; Wang et al., 2025). Although some methods instead model temporal evolution in the parameter space, this introduces its own difficulties: the parameter space is inherently more abstract and higher-dimensional, and its performance is tightly coupled to the architecture and design choices of the predictive model itself. Furthermore, the dynamics governing how parameters evolve across irregular time steps are typically highly nonlinear and lack clear structural priors, making it unstable and prone to error.

Challenge 2: Data non-stationarity complicates the extraction of dynamic temporal patterns across historical environments. Temporal non-stationarity, manifested as shifting data distributions and evolving feature semantics, undermines the assumption of stable statistical regularities over time (Wang et al., 2023b). As a result, periodicity, seasonality, or recurring temporal structure that are readily captured under stationarity often become obfuscated or highly volatile. This renders traditional temporal modeling techniques often fail to generalize in real-world TDG scenarios (Gama et al., 2014). Moreover, future target domains may prioritize distinct contextual cues drawn from historical environments. For instance, some domains may place greater emphasis on long-term trends and macro-level patterns, while others may be highly sensitive to recent, short-term dynamics. Therefore, a robust TDG model should not only adapt to the continuous evolution of temporal patterns but also possess the flexibility to generalize across heterogeneous domain structures.

To address these challenges, we introduce **AdaSpecK**, a spectral-Koopman framework with adaptive context extraction for temporal domain generalization. To mitigate noise fitting to irregularly sampled training domains, we introduce spectral-regularized Koopman dynamics modeling. It applies spectral-aware filtering in the latent space to extract denoised low-frequency trajectories. By focusing on robust, slowly evolving signals, the model enhances resilience to irregular domain shifts and improves generalization to OOD scenarios. It then leverages a Koopman-style transition to model future time evolution and optimizes a Koopman operator that captures system dynamics in a linearized space. To address the difficulty of extracting historical information under non-stationarity, we design a context-informed heterogeneous pattern extraction mechanism. Specifically, we employ a target-conditioned attention module, which enables each future step to attend to distinct subsets of the historical windows. This effectively produces a dynamic, target-specific historical summary of past observations that adapts to the anticipated characteristics of the future domain. Furthermore, by constructing an environmental signature from the current evolutionary pattern, our model adaptively perceives which aspects of the past context are most informative for future prediction via a lightweight, learnable router. Overall, our main contributions are summarized as follows:

- We introduce spectral-regularized Koopman dynamics modeling, which applies spectral-aware filtering in a latent space to extract denoised low-frequency trajectories and learn a Koopman operator to model the system dynamics.

- We propose a context-informed heterogeneous pattern extraction mechanism that leverages target-conditioned attention to exploit the most relevant past for each prediction step, enabling the model to selectively integrate historical contexts through adaptive expert routing.
- Extensive experiments on eight diverse benchmarks that span five classification and three regression tasks show that AdaSpecK achieves state-of-the-art performance.

2 Related Works

Temporal Domain Generalization. Domain Generalization (DG) aims to learn from multiple source domains and generalize to unseen target domains (Muandet et al., 2013; Motiian et al., 2017; Li et al., 2017; Balaji et al., 2018; Dou et al., 2019; Yu et al., 2022; Gulrajani & Lopez-Paz, 2021). DG methods can be classified into three strategies (Wang et al., 2023a): data manipulation (Liu et al., 2018; Qiao et al., 2020), representation learning (Gong et al., 2019; Li et al., 2018b), and learning strategies such as meta-learning (Cai et al., 2023; Li et al., 2018a) and ensembles (Mancini et al., 2018). Temporal Domain Generalization (TDG) extends DG to time-evolving environments, where domain distributions shift over time. Early efforts primarily adopt data-driven strategies. S-MLDG (Li et al., 2020) frames TDG as a sequential meta-learning problem, while TKNets (Zeng et al., 2024) capture temporal dynamics by minimizing the divergence between predicted and actual future domain distributions for classification. Other methods focus on adapting model parameters to evolving domains. Gradient Interpolation (Nasery et al., 2021) extrapolates model updates by supervising the first-order Taylor expansion of the learned function. LSSAE (Qin et al., 2022) introduces a probabilistic latent space to uncover structural regularities in predictive models, and DRAIN (Bai et al., 2023a) uses a recurrent architecture to generate time-adaptive parameters. More recently, TDG has been formalized as a continuous-time setting (Cai et al., 2024), with emerging work exploring frequency-domain analysis of parameter trajectories (Yu et al., 2025) to better characterize temporal concept drift (Gama et al., 2014). Despite this progress, existing methods remain limited: those operating in the data space are often prone to overfitting to domain-specific noise, while those in the parameter space tend to become overly complex and less interpretable. Moreover, none explicitly model dynamic historical environments under non-stationary conditions, overlooking how past contextual information should be adaptively leveraged for future prediction.

Continuous Dynamics and Frequency Learning. Continuous Dynamical Systems (CDS) model temporal evolution via differential equations, offering a natural framework for systems that evolve without discrete time steps. Neural extensions such as Neural ODEs (Chen et al., 2018b) and Latent ODEs (Rubanova et al., 2019) enable flexible, continuous-time representations in deep learning. In parallel, frequency-based learning methods analyze non-stationary temporal patterns by transforming signals into the spectral domain (Wu et al., 2021; 2023), uncovering periodic structures and multi-scale trends that remain obscured in raw time-domain data. For instance, prior works (Zhou et al., 2022; Ye et al., 2024; Yi et al., 2024) leverage spectral decomposition to explicitly disentangle temporal signals into low-frequency structures and high-frequency residuals or noise, and model these components separately to enhance both robustness and predictive performance. Despite their strengths, CDS-based approaches often overlook distribution shifts and structured contextual information, while spectral methods typically lack explicit dynamic modeling. These limitations motivate our core insight: integrating evolving dynamics with frequency-aware representations offers a promising pathway toward improved temporal generalization.

3 Problem Definition

Temporal Domain Generalization (TDG) considers settings in which data distributions evolve over time, with the goal of generalizing to unseen future domains. During training, we are given a sequence of observed domains $\{D(t_1), D(t_2), \ldots, D(t_T)\}$, collected at arbitrary timestamps $T = \{t_1, t_2, \ldots, t_T\}$, where $t_i \in \mathbb{R}^+$ and $t_1 < t_2 < \ldots < t_T$. Each domain at timestamp t consists of N_t input-label pairs, $\mathcal{D}_t = \{(\mathbf{x}_t^{(i)}, y_t^{(i)})\}_{i=1}^{N_t}$, where $(\mathbf{x}_t^{(i)}, y_t^{(i)}) \in \mathcal{X}_t \times \mathcal{Y}_t$ and $\mathcal{X}_t$ and $\mathcal{Y}_t$ denote the input and label spaces at timestamp t. We assume that each domain is sampled from a time-dependent distribution $P_t(X, Y)$, and that the observed sequence undergoes *distribution drift* over time. The objective is to leverage the observed domains to learn a predictive model $f(\cdot; \theta)$ that generalizes effectively to future, unseen OOD target domains associated with timestamps $t' > t_T$.

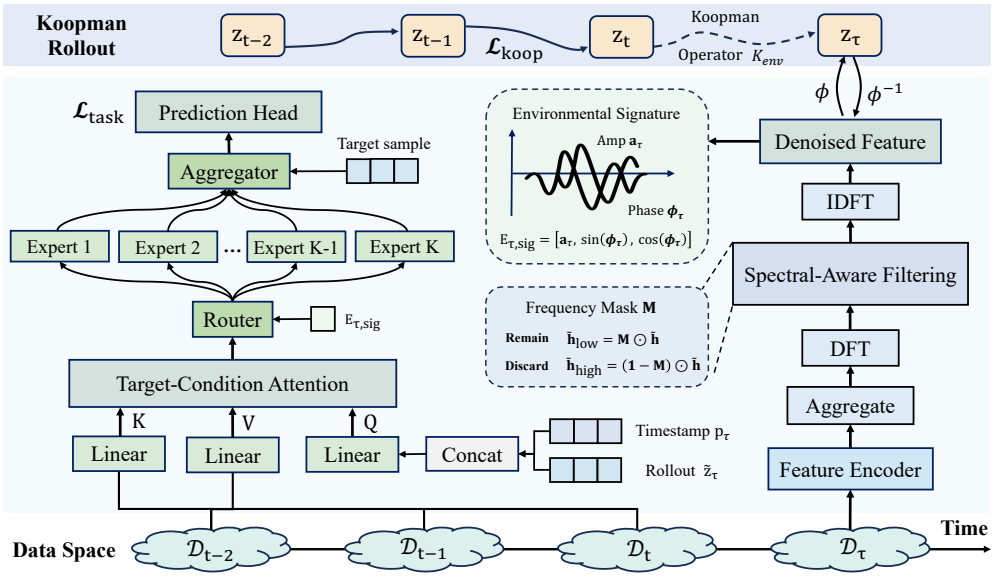


Figure 2: The overview of AdaSpecK.

4 METHODOLOGY

We present **AdaSpecK**, a spectral-Koopman framework with adaptive context extraction for TDG (Figure 2). First, it employs spectral-regularized Koopman dynamics modeling to mitigate domain-specific noise fitting. By applying spectral-aware filtering in a latent space, it extracts denoised, low-frequency trajectories and learns a compact Koopman operator to model future evolution in a robust, linearized space. Second, it incorporates a context-informed heterogeneous pattern extraction mechanism to model complex historical environments under non-stationary. By employing a target-conditioned attention, the model generates dynamic, task-specific historical summaries that attend to distinct subsets of the historical windows. Furthermore, a learned router utilizes environmental signatures to adaptively select the most informative historical information for future prediction.

4.1 SPECTRAL-REGULARIZED KOOPMAN DYNAMICS MODELING

Spectral-Aware Filtering in Latent Space. Direct data-space modeling in TDG often amplifies domain-specific artifacts and noise due to sparse and irregular observation timestamps. To better capture domain structure, we first map the raw observations into a shared latent space. For each domain $\mathcal{D}_t = \{(\mathbf{x}_t^{(i)}, y_t^{(i)})\}_{i=1}^{N_t}$ observed at timestamp t, we encode all samples within this domain using a shared feature encoder $f_\psi : \mathcal{X}_t \to \mathbb{R}^d$ and aggregate them into a domain-level representation that captures coherent domain structure while remaining robust to sample-level variability:

$$h_t = \mathrm{Aggregate}\big(\{f_\psi(\mathbf{x}_t^{(i)})\}_{i=1}^{N_t}\big), \quad t = 1, \ldots, T_c, \tag{1}$$

where $\mathrm{Aggregate}(\cdot)$ denotes domain-level pooling and T_c is the number of historical timestamps, yielding a latent trajectory $\mathbf{h}_{1:T_c} = [h_1, h_2, \ldots, h_{T_c}]^\top \in \mathbb{R}^{T_c \times d}$. While the latent space provides a compact representation, the trajectory remains vulnerable to transient fluctuations that may cause the model to overfit to short-term temporal irregularities. Inspired by frequency-domain insights that system-level temporal trends predominantly reside in low-frequency components (Zhou et al., 2022; Ye et al., 2024), we transfer $\mathbf{h}_{1:T_c}$ to frequency domain via a Discrete Fourier Transform (DFT):

$$\tilde{\mathbf{h}} = \mathcal{F}(\mathbf{h}_{1:T_c}) \in \mathbb{C}^{T_c \times d}, \tag{2}$$

where $\mathcal{F}(\cdot)$ denotes the DFT operator applied across the temporal dimension. To disentangle low- and high-frequency components, we introduce a spectral ratio β to control the proportion of preserved frequency modes. Let $k = \lfloor \beta T_c \rfloor$ denote the number of preserved frequency bins. We construct a binary mask $\mathbf{M} \in \{0, 1\}^{T_c}$ that selects the k lowest-magnitude frequencies:

$$\mathbf{M}[i] = \begin{cases} 1, & \text{if } i \in \mathcal{K}_\beta, \\ 0, & \text{otherwise}, \end{cases} \tag{3}$$

where $\mathcal{K}_\beta$ is the index set corresponding to the k frequencies with the smallest magnitudes. The filtered spectrum is then obtained by applying this mask via the Hadamard product $\odot$:

$$\tilde{\mathbf{h}}_{\text{low}} = \mathbf{M} \odot \tilde{\mathbf{h}}, \quad \tilde{\mathbf{h}}_{\text{high}} = (\mathbf{1} - \mathbf{M}) \odot \tilde{\mathbf{h}}, \tag{4}$$

Finally, applying the inverse DFT to the low-frequency spectrum yields the denoised trajectory:

$$\mathbf{h}_{1:T_c,\text{low}} = \mathcal{F}^{-1}(\tilde{\mathbf{h}}_{\text{low}}) \in \mathbb{R}^{T_c \times d}. \tag{5}$$

By discarding high-frequency components that are often associated with sampling noise or domain-specific perturbations, we obtain a smooth, globally coherent representation that better reflects the intrinsic temporal dynamics of the system. This spectrally filtered latent trajectory enables subsequent structured dynamics modeling to focus on robust, slowly-evolving signals, thereby enhancing resilience to irregular domain shifts and improving generalization to future OOD domains.

Dynamics Modeling via Koopman Operator. Conventional temporal dynamics models, such as recurrent neural networks or models based on ordinary differential equation (ODE), often struggle with highly nonlinear systems or suffer from numerical instability during long-horizon rollouts (Cai et al., 2024). Koopman theory offers a powerful alternative by asserting that nonlinear dynamics can be represented as a linear system in a sufficiently high-dimensional function space (Koopman, 1931; Brunton et al., 2022). Building on this insight, we encode the denoised trajectory $\mathbf{h}_{1:T_c,\text{low}}$ into the Koopman space via a learnable encoder ϕ, leverage a Koopman-style transition to model their future time evolution, and decode back to the latent space via a decoder ϕ^{-1}:

$$z_{t,\text{low}} = \phi(h_{t,\text{low}}), \quad \hat{z}_{t+1,\text{low}} = \mathbf{K}_{\text{env}} z_{t,\text{low}}, \quad \hat{h}_{t+1,\text{low}} = \phi^{-1}(\hat{z}_{t+1,\text{low}}), \tag{6}$$

where $\mathbf{K}_{\text{env}}$ is a trainable Koopman operator that captures the intrinsic system dynamics in the Koopman space. To ensure the Koopman operator accurately captures the underlying dynamics and maintains long-term stability, we employ a multi-faceted loss function composed of three complementary terms. First, we adopt the Koopman transition loss to enforce local fidelity of the dynamics in the Koopman space by penalizing one-step prediction errors:

$$\mathcal{L}_{\text{koop}} = \sum_{t=1}^{T_c-1} \|z_{t+1,\text{low}} - \hat{z}_{t+1,\text{low}}\|_2^2, \tag{7}$$

To guarantee that the Koopman-evolved states remain grounded in the original feature manifold, we supervise the decoded rollout in the hidden space against the ground-truth hidden states:

$$\mathcal{L}_{\text{rec}} = \sum_{t=1}^{T_c-1} \left\| h_{t+1,\text{low}} - \hat{h}_{t+1,\text{low}} \right\|_2^2, \tag{8}$$

To mitigate error accumulation over extended horizons, we introduce a consistency constraint that aligns the ground-truth trajectories with the dynamic evolution in the latent space:

$$\mathcal{L}_{\text{consis}} = \sum_{i=1}^{T_c} \sum_{j=1}^{i} \left\| h_{i,\text{low}} - h_{i,\text{low}}^{j \to i} \right\|_2, \tag{9}$$

where $h_{i,\text{low}}^{j \to i}$ denotes the latent state at step i propagated from the j-th anchor. This encourages the model to learn a coherent and path-independent dynamical trajectory, thereby improving robustness to temporal shifts and cumulative drift. Overall, the three losses play different but complementary roles, encouraging the latent dynamics to be both locally accurate and globally stable.

4.2 Context-Informed Heterogeneous Pattern Extraction

Target-Conditioned Historical Summary. Data in TDG tasks often exhibit non-stationarity, as the underlying patterns evolve over time. This temporal drift complicates the modeling of dynamic temporal dependencies across shifting historical contexts and renders static, global representations inadequate for accurate long-term forecasting. To effectively capture diverse temporal dynamics and obtain robust historical patterns, we first aggregate domain-level representations $\mathbf{h}_{1:T_c}$ from past domains into a unified contextual embedding E_{ctx}:

$$E_{\text{ctx}} = g_{\text{ctx}}\big(\text{concat}[h_1, h_2, \ldots, h_{T_c}]\big), \tag{10}$$

where g_{ctx} is a lightweight multilayer perceptron (MLP). Crucially, instead of relying on a fixed summary of the entire history, we introduce a target-conditioned attention mechanism. For each future prediction step $\tau \in \{1, \ldots, H\}$, we construct a dynamic query that encapsulates its anticipated state and its absolute timestep:

$$E_{\tau,q} = \text{concat}\big[\hat{z}_\tau,\, p_\tau\big], \tag{11}$$

where $\hat{z}_\tau$ is obtained via rollout using the Koopman operator $\mathbf{K}_{\text{env}}$, and p_τ is a positional embedding that provides explicit temporal grounding. We then compute cross-attention between this query $E_{\tau,q}$ and the historical contextual embedding E_{ctx}:

$$E_{\text{history}} = \text{softmax}\left(E_{\tau,q} W_Q \left(E_{\text{ctx}} W_K\right)^T / \sqrt{d_k}\right) E_{\text{ctx}} W_V \tag{12}$$

where $W_Q \in \mathbb{R}^{d\times d}$, $W_K \in \mathbb{R}^{d\times d}$, $W_V \in \mathbb{R}^{d\times d}$ are trainable projection matrices, and d_k denotes the key dimension used for scaling. This mechanism enables the model to selectively attend to semantically and temporally relevant segments of the historical context conditioned on each future step, rather than relying on a uniform historical abstraction. Consequently, the resulting representation E_{history} serves as a dynamic, target-specific historical summary, which better captures temporal evolution patterns and provides more precise contextual support for subsequent prediction.

Adaptive Perception via Environmental Signature. Although the historical summary provides a dynamic view of relevant past contexts, future target domains may still emphasize distinct contextual cues from historical environments. For example, some target domains may be primarily dominated by pronounced weekly cycles, whereas others may be more sensitive to recent transient events or seasonal variations. This discrepancy indicates that a universal historical encoding is insufficient for capturing heterogeneous temporal pattern extraction. Instead, the model should be able to adaptively reinterpret historical information according to the temporal characteristics of the emerging target environment. To facilitate such adaptive perception, we first construct a compact environmental signature from the predicted future trajectory. Specifically, for each future step $\tau \in \{1, \ldots, H\}$, we derive its low-frequency representation $\mathbf{h}_{\tau,\text{low}}$ following Eqs.(1)-(5).

While temporal distributional shifts typically do not alter the physical mechanism of the system, they do break stable statistical regularity over time. As a result, periodicity, seasonality, and other temporal structures that are readily captured under stationarity may become obscured or highly volatile. To identify relatively stable structural cues under such shifts, we extract two key features from the low-frequency components: amplitude $\mathbf{a}_\tau = \text{Amp}(\mathbf{h}_{\tau,\text{low}})$, which reflects the strength of each periodic mode, and phase $\boldsymbol{\phi}_\tau = \text{Phase}(\mathbf{h}_{\tau,\text{low}})$, which encodes temporal alignment and structural sequencing. To ensure differentiability and avoid angular discontinuities, phase is represented via sine and cosine projections. The resulting environmental signature at time τ is defined as:

$$\text{E}_{\tau,\text{sig}} = \big[\mathbf{a}_\tau,\, \sin(\boldsymbol{\phi}_\tau),\, \cos(\boldsymbol{\phi}_\tau)\big]. \tag{13}$$

The environmental signature is then fed to an MLP-based router network g_{route}:

$$\boldsymbol{w} = g_{\text{route}}\big(\text{E}_{\tau,\text{sig}}\big), \tag{14}$$

where $\boldsymbol{w} \in \mathbb{R}^K$ denotes Softmax-normalized expert weights. The target-conditioned historical summary E_{history} is then routed via K experts and aggregated into a adaptive pattern representation:

$$E_{\text{pattern}} = \sum_{k=1}^{K} \boldsymbol{w}_k g_{\text{expert,k}}(E_{\text{history}}), \tag{15}$$

To encourage experts to learn distinct and non-redundant temporal patterns, we introduce a semantic orthogonality loss $\mathcal{L}_{\text{sem}}$. This loss encourages the routing weight matrix W_{gate} to maintain diverse semantic directions, preventing expert collapse where all experts converge to the same mean pattern:

$$W = W_{\text{gate}}, \qquad K = \text{normalize}(W), \qquad G = KK^\top, \qquad \mathcal{L}_{\text{sem}} = \text{MSE}(G, I), \tag{16}$$

where W_{gate} is the final-layer weight matrix of the router and I is the identity matrix. The prediction head then fuses features of target sample E_{target} with E_{pattern} to yield the final prediction:

$$\hat{y} = g_{\text{head}}([E_{\text{target}}, E_{\text{pattern}}]). \tag{17}$$

Finally, the model is optimized using a composite objective comprising the task-specific loss $\mathcal{L}_{\text{task}}$ with multiple regularization terms:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \lambda_K \big(\mathcal{L}_{\text{koop}} + \mathcal{L}_{\text{rec}}\big) + \lambda_S \mathcal{L}_{\text{sem}} + \lambda_C \mathcal{L}_{\text{consis}}. \tag{18}$$

where $\mathcal{L}_{\text{task}} = \sum_{i=1}^{N} \ell(y_i, \hat{y}_i)$, using cross-entropy for classification and mean absolute error (MAE) for regression, and λ_K, λ_S, λ_C are hyperparameters controlling the relative contributions.

Table 1: Performance comparison on eight TDG datasets. Classification results are reported as error rates (%) except for Twitter (AUC); regression results report MAE. 'N/A' denotes unsupported tasks. The best and second-best results are in bold and underlined. ↑: higher is better, ↓: lower is better.

Model	Classification					Regression		
	Rot-MNIST ↓	Twitter ↑	Yearbook ↓	ONP ↓	Shuttle ↓	Cyclone ↓	House ↓	Appliance ↓
Offline	6.6 ± 0.2	0.54 ± 0.09	8.6 ± 1.0	33.8 ± 0.6	0.77 ± 0.10	18.7 ± 1.4	19.9 ± 0.1	10.2 ± 1.1
LastDomain	74.2 ± 0.9	0.54 ± 0.12	11.3 ± 1.3	36.0 ± 0.2	0.91 ± 0.18	22.3 ± 0.7	20.6 ± 0.7	9.1 ± 0.7
IncFinetune	57.1 ± 1.4	0.52 ± 0.01	11.0 ± 0.8	34.0 ± 0.3	0.83 ± 0.07	19.9 ± 0.7	20.6 ± 0.2	8.9 ± 0.5
IRM	8.6 ± 0.4	0.53 ± 0.11	8.3 ± 0.5	45.1 ± 0.1	0.88 ± 0.08	18.0 ± 0.8	19.8 ± 0.2	5.5 ± 0.1
V-REx	8.6 ± 0.3	0.58 ± 0.05	8.9 ± 0.5	33.8 ± 0.1	1.90 ± 0.21	17.7 ± 0.5	20.2 ± 0.1	4.5 ± 0.1
CIDA	8.3 ± 0.7	0.63 ± 0.03	8.4 ± 0.8	34.7 ± 0.6	7.71 ± 0.12	17.0 ± 0.4	10.2 ± 1.0	8.7 ± 0.2
TKNets	37.7 ± 2.0	0.57 ± 0.04	8.4 ± 0.3	33.7 ± 0.2	0.94 ± 0.28	N/A	N/A	N/A
DRAIN	59.1 ± 2.3	0.57 ± 0.04	10.5 ± 1.0	38.3 ± 1.2	0.26 ± 0.05	23.6 ± 0.5	9.8 ± 0.1	6.4 ± 0.4
DRAIN-Δt	57.2 ± 1.8	0.59 ± 0.02	11.0 ± 1.2	38.5 ± 1.1	0.25 ± 0.05	26.2 ± 4.6	9.9 ± 0.1	6.5 ± 0.4
DeepODE	48.6 ± 3.2	0.64 ± 0.02	8.8 ± 2.2	33.6 ± 0.2	2.88 ± 0.71	18.5 ± 3.3	10.7 ± 0.4	$\underline{4.5 \pm 0.1}$
Koodos	4.8 ± 0.3	0.69 ± 0.02	$\underline{5.9 \pm 0.7}$	$\underline{33.5 \pm 0.4}$	0.24 ± 0.04	$\underline{16.1 \pm 1.8}$	$\underline{9.0 \pm 0.4}$	4.8 ± 0.3
Frekoo	$\underline{4.7 \pm 0.2}$	$\underline{0.71 \pm 0.02}$	13.3 ± 0.8	34.2 ± 0.3	$\underline{0.20 \pm 0.02}$	19.5 ± 0.2	$\underline{9.0 \pm 0.1}$	5.4 ± 1.1
AdaSpecK	$\mathbf{4.5 \pm 0.2}$	$\mathbf{0.74 \pm 0.02}$	$\mathbf{4.5 \pm 0.5}$	$\mathbf{33.3 \pm 0.2}$	$\mathbf{0.18 \pm 0.02}$	$\mathbf{9.3 \pm 0.8}$	$\mathbf{8.9 \pm 0.3}$	$\mathbf{4.3 \pm 0.1}$

5 Experiments

5.1 Experiment Settings

Datasets. Following Koodos (Cai et al., 2024) and FreKoo (Yu et al., 2025), we evaluate AdaSpecK on eight datasets spanning both classification and regression tasks. The five classification datasets include Rotated MNIST (Rot-MNIST) (Deng, 2012), Twitter Influenza Risk (Twitter) (Zhao et al., 2017; Bai et al., 2023b), and YearBook (Yao et al., 2022), Online News Popularity (ONP) (Fernandes et al., 2015), and Shuttle (UCI Machine Learning Repository, 2025); the three regression datasets consist of Tropical Cyclone Intensity (Cyclone) (Chen et al., 2018a), House Prices (House) (Cock, 2011), and ApplianceEnergy (Appliance) (Candanedo et al., 2017). These datasets capture diverse forms of temporal domain shift and real-world non-stationarity, providing a comprehensive benchmark for evaluation. More details of the datasets are provided in Appendix A.1.

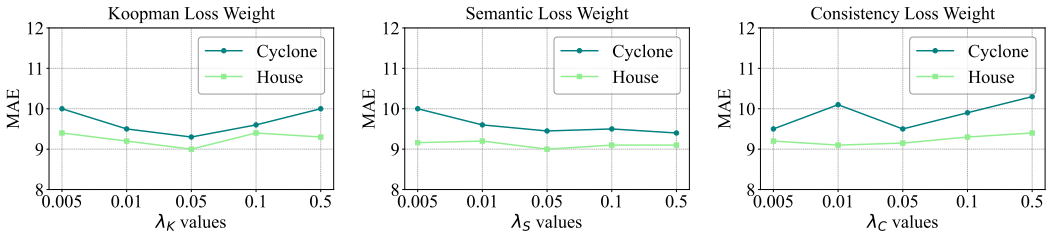
Baselines. We compare our method against baselines from three representative categories: (1) Time-agnostic methods, including Offline (Cai et al., 2024; Yu et al., 2025), LastDomain (Cai et al., 2024; Yu et al., 2025), IncFinetune (Cai et al., 2024; Yu et al., 2025), IRM (Arjovsky et al., 2019), and V-REx (Krueger et al., 2021); (2) Discrete temporal domain adaptation/generalization methods, including CIDA (Wang et al., 2020), TKNets (Zeng et al., 2024), DRAIN, and DRAIN-Δt (Bai et al., 2023a); and (3) Continuous temporal domain generalization methods, including DeepODE (Chen et al., 2018b; Rubanova et al., 2019), Koodos (Cai et al., 2024) and Frekoo (Yu et al., 2025). For any baseline that does not support a particular dataset, we report its result as N/A in Table 1.

Implementation details. AdaSpecK is implemented in PyTorch and trained on an NVIDIA GeForce RTX 3090 GPU. We use the Adam optimizer with cosine annealing (learning rate of 5×10^{-4}, weight decay of 0.005). We set the feature dimension $d = 32$, the low-frequency ratio $\beta = 0.40$. The expert count K is set to 4 for Twitter, House, ONP, and Appliance, and 8 for Rotated MNIST, YearBook, Cyclone, and Shuttle. Additional details are provided in the Appendix A.3.

Metrics. For classification tasks, we report the error rate (%), which measures the proportion of incorrectly predicted labels. Due to severe label imbalance in the Twitter dataset, we instead use the Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC) curve. For regression tasks, performance is measured by Mean Absolute Error (MAE). All models are trained on training domains and then evaluated on unseen test domains to assess their generalization capability across temporal shifts. To ensure statistical reliability, each method is run five times with different random seeds, and we report the mean and standard deviation of the evaluation metrics across these runs.

5.2 Experimental Results

Main Results. Table 1 shows that AdaSpecK achieves the best performance across all eight benchmarks, highlighting its superior capability in handling non-stationary temporal shifts. Time-agnostic



(a) Sensitivity of λ_K (b) Sensitivity of λ_S (c) Sensitivity of λ_C

Figure 3: Sensitivity analysis of loss weight λ_K, λ_S, λ_C.

baselines (e.g., Offline, LastDomain, and IncFinetune) suffer significant performance degradation on long-horizon datasets, confirming that naive reuse of historical data is inadequate under evolving temporal drift. Compared to state-of-the-art continuous TDG baselines (e.g., Koodos and Frekoo), AdaSpecK demonstrates consistent superiority. Notably, our method achieves the most substantial gains on Twitter, YearBook, Cyclone, and Appliance, where the datasets are characterized by multi-step future evolution. This resilience suggests that our learned latent dynamics provide a more stable and robust representation of temporal evolution than approaches relying solely on short-range interpolation or adjacent-domain transitions, avoiding overfitting to domain-specific artifacts or local noise. Moreover, AdaSpecK explicitly models evolving historical contexts through a context-informed, heterogeneous pattern extraction mechanism, adaptively determining how past information should be leveraged for future prediction. In summary, the proposed spectral-Koopman framework with adaptive context extraction not only improves average predictive accuracy but also enhances robustness and stability under persistent temporal distribution shifts.

Ablation Study. Table 2 validates the necessity of each proposed component. Regarding architecture, modeling dynamics directly in latent space without Koopman (w/o Koop.) degrades performance, confirming its role in enabling effective learning by mapping dynamics into a robust, linearized subspace. Removing spectral-aware filtering (w/o Spec.) also harms results, highlighting the importance of frequency separation for isolating slowly-varying structures critical to temporal extrapolation. Ablating dynamic target-conditioned historical summary (w/o E_{history}) or environmental signatures in the context router (w/o $\mathrm{E}_{\tau,\text{sig}}$) both lead to suboptimal results, underscoring the value of adaptive historical modeling and environment-aware routing. Loss ablations further demonstrate the importance of each objective, with the full loss integration yielding the best results.

Table 2: Ablation study of AdaSpecK. The best and second-best results are in bold and underlined.

Variants	Classification			Regression	
	Rot-MNIST ↓	Twitter ↑	YearBook ↓	Cyclone ↓	House ↓
w/o Koop.	4.8	0.67	<u>4.8</u>	10.8	9.1
w/o Spec.	4.9	0.63	<u>4.8</u>	10.6	9.0
w/o E_{history}	4.8	<u>0.70</u>	4.9	10.5	9.2
w/o $\mathrm{E}_{\tau,\text{sig}}$	<u>4.7</u>	0.68	5.3	<u>9.6</u>	**8.9**
w/o $\mathcal{L}_{\text{koop}}$	5.1	<u>0.69</u>	6.2	<u>9.9</u>	9.0
w/o $\mathcal{L}_{\text{rec}}$	<u>4.9</u>	0.69	<u>5.2</u>	10.9	9.0
w/o $\mathcal{L}_{\text{sem}}$	5.0	0.65	5.3	10.5	9.1
w/o $\mathcal{L}_{\text{consis}}$	8.4	0.66	5.4	10.5	<u>9.0</u>
AdaSpecK	**4.5 ± 0.2**	**0.74 ± 0.02**	**4.5 ± 0.5**	**9.3 ± 0.8**	**8.9 ± 0.3**

5.3 Further Analysis

Sensitivity Analysis. We examine the impact of loss weights λ_K, λ_S, λ_C on diverse datasets (Cyclone, House, Appliance). As shown in Figure 3, a consistent pattern is that moderate coefficients lead to the best or near-best performance, whereas overly large coefficients slightly degrade accuracy, especially on Cyclone. This confirms that the auxiliary objectives serve as effective structural regularizers, with optimal results achieved under balanced weighting.

Decision Boundary Visualization. Figure 4 compares the decision regions of AdaSpecK, Koodos, and DeepODE on the YearBook test domains. Visually, AdaSpecK produces smoother boundaries that better reflect the underlying class separation while adapting to domain-specific shifts. In contrast, Koodos shows locally distorted or biased toward one cluster, and DeepODE fails to track evolving class geometry. The per-domain error annotations further show that AdaSpecK maintains consistent performance across target domains. Together, these findings demonstrate that AdaSpecK's integration of spectral-regularized Koopman dynamics modeling and target-conditioned context extraction effectively captures temporally evolving decision structures.

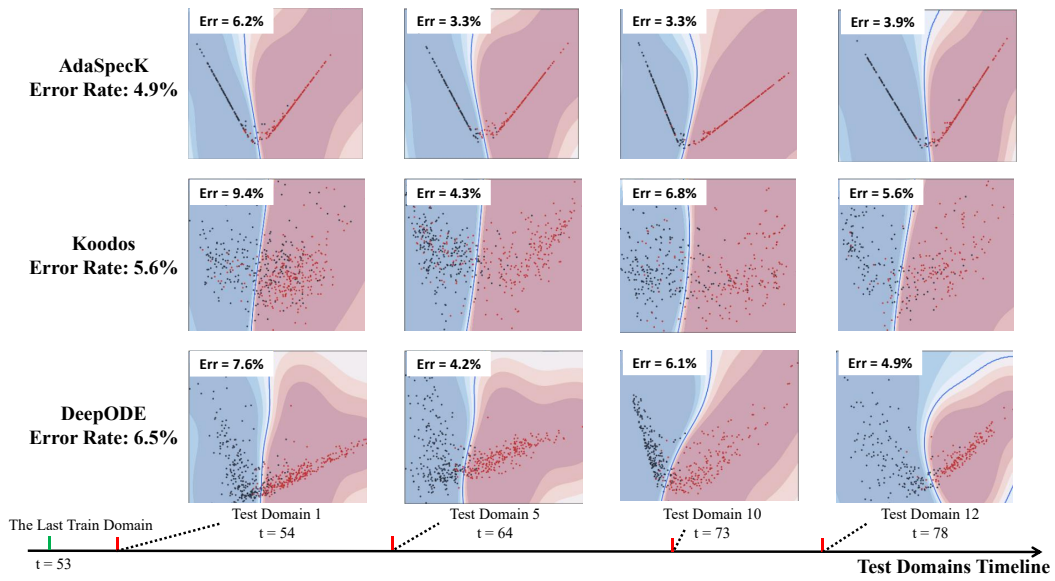
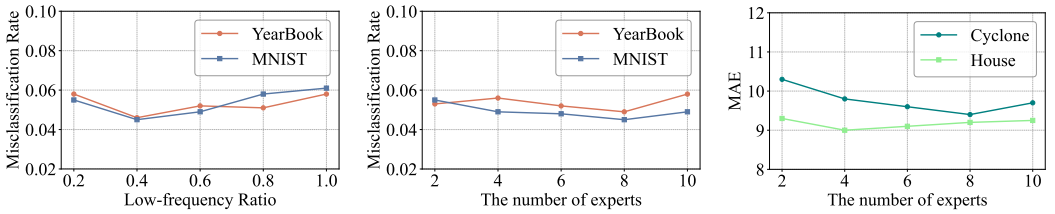


Figure 4: Visualization of decision boundaries on YearBook. Blue and red dots represent different data classes. Rows: our method vs. baselines; columns: different test domains over time.



(a) Impact of Spectral Ratio (b) MNIST & YearBook (c) Cyclone & House

Figure 5: (a) Impact of the spectral ratio β; (b, c) Choice of the number of experts.

Impact of Spectral ratio. Figure 5a illustrates the effect of the spectral cutoff ratio β. We observe that performance is suboptimal with an excessively small β, while a moderate ratio yields peak performance. However, further increasing β leads to a noticeable decline in accuracy. This trend aligns with the design principles of spectral decomposition: the low-frequency branch must preserve denoised slowly-varying temporal structures conducive to Koopman dynamic modeling, while preventing the inclusion of high-frequency noise that could impair model stability.

Choice of Expert Numbers. We study the impact of expert number $K \in [2, 4, 6, 8, 10]$ (Figure 5b and 5c). A small K fails to adequately capture the diversity of heterogeneous historical patterns, while a large K increases computational cost and complexity, potentially leading to overfitting and degraded results. Overall, K in the range of 4-8 offers the best balance and is ideal for practical use.

6 Conclusion

This paper proposes **AdaSpecK**, a spectral-Koopman framework with adaptive context extraction for temporal domain generalization. To mitigate noise fitting to irregular sampling, we introduce spectral-regularized Koopman dynamics modeling. It applies spectral-aware filtering in a latent space to extract denoised low-frequency trajectories and learns a compact Koopman operator to model system dynamics in a robust, linearized space. To handle non-stationarity in historical contexts, we design a context-informed heterogeneous pattern extraction mechanism. A target-conditioned attention module enables each future timestep to attend to distinct subsets of the historical windows, while an environmental signature derived from the current evolutionary pattern guides a learned router to adaptively identify the most predictive past context. Extensive experiments across eight classification and regression benchmarks show that AdaSpecK achieves SOTA performance.

AI USE STATEMENT

In this work, we used generative AI tools to improve the clarity, grammar, and overall quality of the writing. We have not used generative AI tools for generating the core ideas, formulating the research questions, and developing the methodology. We have reviewed and edited all AI-assisted work. In particular, we examined the correctness, logical coherence, and contextual appropriateness of any AI-generated text to ensure that it accurately reflected our intended meaning and met academic standards. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work is conducted entirely on publicly available benchmark datasets, as detailed in the paper, and does not involve the release of any personal or sensitive information. No human subjects are involved in this research, ensuring that our work complies with ethical standards in research integrity.

REPRODUCIBILITY STATEMENT

All experiments in this work are conducted on publicly available benchmark datasets, and the reported performance of AdaSpecK is fully reproducible. Details of model architecture, training procedures, and evaluation protocols are provided in the main text and appendix. To further facilitate reproducibility, we release the code and datasets at https://anonymous.4open.science/r/Ada-Spec-K.

REFERENCES

Martín Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *CoRR*, abs/1907.02893, 2019.

Guangji Bai, Chen Ling, and Liang Zhao. Temporal domain generalization with drift-aware dynamic neural networks. In *ICLR*, 2023a.

Guangji Bai, Johnny Torres, Junxiang Wang, Liang Zhao, Cristina L. Abad, and Carmen Vaca. Sign-regularized multi-task learning. In *Proceedings of the 2023 SIAM International Conference on Data Mining, SDM*, pp. 793–801. SIAM, 2023b.

Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. In *NeurIPS*, pp. 1006–1016, 2018.

Steven L. Brunton, Marko Budisic, Eurika Kaiser, and J. Nathan Kutz. Modern koopman theory for dynamical systems. *SIAM Rev.*, 64(2):229–340, 2022.

Zekun Cai, Renhe Jiang, Xinyu Yang, Zhaonan Wang, Diansheng Guo, Hill Hiroki Kobayashi, Xuan Song, and Ryosuke Shibasaki. Memda: Forecasting urban time series with memory-based drift adaptation. In *CIKM*, pp. 193–202, 2023.

Zekun Cai, Guangji Bai, Renhe Jiang, Xuan Song, and Liang Zhao. Continuous temporal domain generalization. In *NeurIPS*, 2024.

Luis M. Candanedo, Véronique Feldheim, and Dominique Deramaix. Data driven prediction models of energy use of appliances in a low-energy house. *Energy and Buildings*, 140:81–97, 2017. doi: 10.1016/j.enbuild.2017.01.083.

Boyo Chen, Buo-Fu Chen, and Hsuan-Tien Lin. Rotation-blended cnns on a new open dataset for tropical cyclone image-to-intensity regression. In *SIGKDD*, pp. 90–99. ACM, 2018a.

Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *NeurIPS 2018*, pp. 6572–6583, 2018b.

Dean De Cock. Ames, iowa: Alternative to the boston housing data as an end of semester regression project. *Journal of Statistics Education*, 19(3), 2011. doi: 10.1080/10691898.2011.11889627.

Li Deng. The MNIST database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Process. Mag.*, 29(6):141–142, 2012.

Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. In *NeurIPS*, pp. 6447–6458, 2019.

Kelwin Fernandes, Pedro Vinagre, and Paulo Cortez. A proactive intelligent decision support system for predicting the popularity of online news. In *Progress in Artificial Intelligence*, volume 9273 of *Lecture Notes in Computer Science*, pp. 535–546. Springer, 2015. doi: 10.1007/978-3-319-23485-4_53.

João Gama, Indre Zliobaite, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM Comput. Surv.*, 46(4):44:1–44:37, 2014.

Rui Gong, Wen Li, Yuhua Chen, and Luc Van Gool. DLOW: domain flow for adaptation and generalization. In *CVPR*, pp. 2477–2486. Computer Vision Foundation / IEEE, 2019.

Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *ICLR*, 2021.

Elise Han, Chengpiao Huang, and Kaizheng Wang. Model assessment and selection under temporal distribution shift. In *ICML*, Proceedings of Machine Learning Research, pp. 17374–17392, 2024.

B. O. Koopman. Hamiltonian systems and transformation in hilbert space. *Proceedings of the National Academy of Sciences*, 17(5):315–318, 1931. doi: 10.1073/pnas.17.5.315.

David Krueger, Ethan Caballero, Jörn-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Rémi Le Priol, and Aaron C. Courville. Out-of-distribution generalization via risk extrapolation (REx). In *ICML*, pp. 5815–5826. PMLR, 2021.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Deeper, broader and artier domain generalization. In *ICCV*, pp. 5543–5551, 2017.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Learning to generalize: Meta-learning for domain generalization. In *AAAI*, pp. 3490–3497, 2018a.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Sequential learning for domain generalization. In Adrien Bartoli and Andrea Fusiello (eds.), *ECCV 2020 Workshops*, Lecture Notes in Computer Science, pp. 603–619. Springer, 2020.

Wen Li, Zheng Xu, Dong Xu, Dengxin Dai, and Luc Van Gool. Domain generalization and adaptation using low rank exemplar svms. *IEEE transactions on pattern analysis and machine intelligence*, 40(5):1114–1127, 2018b.

Alexander H. Liu, Yen-Cheng Liu, Yu-Ying Yeh, and Yu-Chiang Frank Wang. A unified feature disentangler for multi-domain image translation and manipulation. In *NeurIPS*, pp. 2595–2604, 2018.

Jie Lu, Anjin Liu, Fan Dong, Feng Gu, João Gama, and Guangquan Zhang. Learning under concept drift: A review. *IEEE transactions on knowledge and data engineering*, 31(12):2346–2363, 2019.

Massimiliano Mancini, Samuel Rota Bulò, Barbara Caputo, and Elisa Ricci. Best sources forward: Domain generalization through source-specific nets. In *ICIP*, pp. 1353–1357. IEEE, 2018.

Saeid Motiian, Marco Piccirilli, Donald A. Adjeroh, and Gianfranco Doretto. Unified deep supervised domain adaptation and generalization. In *ICCV*, pp. 5716–5726, 2017.

Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *ICML*, pp. 10–18, 2013.

Anshul Nasery, Soumyadeep Thakur, Vihari Piratla, Abir De, and Sunita Sarawagi. Training for the future: A simple gradient interpolation loss to generalize along time. In *NeurIPS*, pp. 19198–19209, 2021.

Fengchun Qiao, Long Zhao, and Xi Peng. Learning to learn single domain generalization. In *CVPR*, pp. 12553–12562. Computer Vision Foundation / IEEE, 2020.

Tiexin Qin, Shiqi Wang, and Haoliang Li. Generalizing to evolving domains with latent structure-aware sequential autoencoder. In *ICML*, Proceedings of Machine Learning Research, pp. 18062–18082. PMLR, 2022.

Yulia Rubanova, Ricky T. Q. Chen, and David Duvenaud. Latent odes for irregularly-sampled time series. *CoRR*, abs/1907.03907, 2019. URL http://arxiv.org/abs/1907.03907.

UCI Machine Learning Repository. Statlog (shuttle) [dataset], 2025. URL https://archive.ics.uci.edu/dataset/148/statlog+shuttle. Accessed: 2026-05-02.

Hao Wang, Hao He, and Dina Katabi. Continuously indexed domain adaptation. In *ICML*, volume 119, pp. 9898–9907. PMLR, 2020.

Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip S. Yu. Generalizing to unseen domains: A survey on domain generalization. *IEEE transactions on knowledge and data engineering*, 35(8):8052–8072, 2023a.

Jingyuan Wang, Zhimei Ren, Ruohan Zhan, and Zhengyuan Zhou. Distributionally robust policy learning under concept drifts. In *ICML*, PMLR, 2025.

Rui Wang, Yihe Dong, Sercan Ö. Arik, and Rose Yu. Koopman neural operator forecaster for time-series with temporal distributional shifts. In *ICLR*, 2023b.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *NeurIPS*, pp. 22419–22430, 2021.

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

Huaxiu Yao, Caroline Choi, Bochuan Cao, Yoonho Lee, Pang Wei Koh, and Chelsea Finn. Wild-time: A benchmark of in-the-wild distribution shift over time. In *NeurIPS*, volume 35, pp. 10309–10324, 2022.

Weiwei Ye, Songgaojun Deng, Qiaosha Zou, and Ning Gui. Frequency adaptive normalization for non-stationary time series forecasting. In *NeurIPS*, 2024.

Kun Yi, Jingru Fei, Qi Zhang, Hui He, Shufeng Hao, Defu Lian, and Wei Fan. Filternet: Harnessing frequency filters for time series forecasting. In *NeurIPS*, 2024.

Dazhou Yu, Guangji Bai, Yun Li, and Liang Zhao. Deep spatial domain generalization. In Xingquan Zhu, Sanjay Ranka, My T. Thai, Takashi Washio, and Xindong Wu (eds.), *ICDM*, pp. 1293–1298. IEEE, 2022.

En Yu, Jie Lu, Xiaoyu Yang, Guangquan Zhang, and Zhen Fang. Learning robust spectral dynamics for temporal domain generalization. *arXiv preprint arXiv:2505.12585*, 2025.

Qiuhao Zeng, Wei Wang, Fan Zhou, Gezheng Xu, Ruizhi Pu, Changjian Shui, Christian Gagné, Shichun Yang, Charles X. Ling, and Boyu Wang. Generalizing across temporal domains with koopman operators. *AAAI*, 38(15):16651–16659, 2024.

Liang Zhao, Qian Sun, Jieping Ye, Feng Chen, Chang-Tien Lu, and Naren Ramakrishnan. Feature constrained multi-task learning models for spatiotemporal event forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 29(5):1059–1072, 2017.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *ICML*, pp. 27268–27286. PMLR, 2022.

A APPENDIX

A.1 DATASETS

Following Koodos (Cai et al., 2024) and FreKoo (Yu et al., 2025), and focusing on the datasets that are both discussed in the two reference papers and implemented in the new codebase, we evaluate AdaSpecK on eight benchmarks covering classification and regression under different temporal drift patterns. Their main characteristics are summarized as follows:

- **Rotated MNIST (Rot-MNIST).** This is a synthetic image classification benchmark built from MNIST digits (Deng, 2012), where concept drift is created by gradually rotating the images across domains as in the temporal benchmark protocols of Koodos (Cai et al., 2024) and FreKoo (Yu et al., 2025). The domains are temporally ordered, and the prediction target always lies in later unseen domains. Its main difficulty comes from a steadily increasing geometric transformation rather than label noise.
- **Twitter.** This dataset uses tweet streams to estimate influenza activity. The domains correspond to temporally separated collection windows during multiple flu seasons, so the drift is driven by changes in tweet volume, seasonal disease activity, and evolving user behavior (Cai et al., 2024; Yu et al., 2025). Compared with synthetic rotation datasets, it reflects a real-world temporally evolving classification problem with stronger non-stationarity.
- **YearBook.** This is a gender classification benchmark built from frontal yearbook portraits collected over many decades, following the YearBook temporal-shift benchmark in Wild-Time (Yao et al., 2022). Each domain corresponds to one sampled year in chronological order. The drift arises from long-term changes in clothing, hairstyle, imaging quality, and broader social norms, making it a representative visual benchmark with slow but persistent temporal shift.
- **Cyclone.** This regression dataset predicts tropical cyclone intensity from satellite imagery. Each domain is associated with temporally ordered cyclone observations, and the shift is induced by seasonal and regional variations in storm appearance and wind intensity (Cai et al., 2024; Yu et al., 2025). It is therefore a visually complex forecasting benchmark with substantial temporal variation.
- **House.** This regression benchmark predicts house prices from tabular features collected over multiple years. The domains are organized chronologically, and the underlying drift mainly reflects changing market conditions and temporal economic variation, making it a real-world nonstationary tabular dataset (Cock, 2011; Cai et al., 2024; Yu et al., 2025).
- **Online News Popularity (ONP).** This classification dataset contains news articles from Mashable with 58 heterogeneous features, and the task is to predict article popularity (Fernandes et al., 2015). The data are split into time-ordered domains over roughly two years, with the first five domains used for training and the last one for testing in the TDG protocol. Its drift is relatively mild compared with the other benchmarks, but it still captures gradual changes in how content attracts social attention over time.
- **Shuttle.** This dataset is a temporally ordered multi-class classification benchmark derived from flight-status data and is known for its strong class imbalance (UCI Machine Learning Repository, 2025; Cai et al., 2024; Yu et al., 2025). The domains are formed chronologically, with earlier periods used for training and the latest period used for testing. The challenge comes from real-world distribution changes in operational patterns over time rather than from manually constructed perturbations.
- **ApplianceEnergy (Appliance).** This regression benchmark predicts appliance energy consumption from indoor sensor readings in a low-energy building (Candanedo et al., 2017). The data are recorded at 10-minute intervals over several months and then partitioned into chronological domains. The drift is associated with changing usage behavior and environmental conditions across time, which makes it a practical benchmark for periodic and real-world temporal variation.

A.2 BASELINE

We compare AdaSpecK with several families of baselines that reflect different ways of handling temporal drift. Offline, LastDomain, and IncFinetune follow the standard non-temporal or sequential training baselines used in recent TDG studies (Cai et al., 2024; Yu et al., 2025), where Offline trains a single model on the union of all source domains and ignores temporal order entirely, LastDomain uses only the most recent observed source domain to test whether short-range recency alone is sufficient, and IncFinetune updates the predictor sequentially from earlier to later domains as a practical continual-training baseline. IRM and V-REx are invariant-learning methods that aim to suppress domain-specific shortcuts by favoring predictors whose risks are stable across source domains (Arjovsky et al., 2019; Krueger et al., 2021), although they do not explicitly model domain evolution over time. CIDA incorporates temporal or domain information as a continuous conditioning variable and aligns representations across domains (Wang et al., 2020), while TKNets models transitions between temporally ordered domains through a dedicated temporal transfer mechanism (Zeng et al., 2024). DRAIN and DRAIN-Δt forecast future predictors by learning parameter dynamics from the observed domain sequence, with the latter additionally incorporating time-gap information for non-uniform intervals (Bai et al., 2023a). DeepODE models temporal evolution in a latent dynamical system defined by neural ordinary differential equations (Chen et al., 2018b; Rubanova et al., 2019), and Koodos introduces Koopman-space parameter evolution with structured dynamic constraints (Cai et al., 2024). We also report FreKoo as a strong recent temporal generalization reference that combines spectral decomposition and Koopman modeling to separate low-frequency temporal structure from high-frequency variations (Yu et al., 2025). Together, these methods cover time-agnostic training, invariant learning, temporal alignment, parameter-forecasting approaches, and frequency-aware temporal dynamics.

A.3 IMPLEMENTATION DETAILS

All results were averaged over 5 independent runs and we report the mean and standard deviation. We implement AdaSpecK in PyTorch and conduct all experiments on a 64-bit machine with an NVIDIA GeForce RTX 3090 GPU. Consistent with the summary in the main text, we use Adam with cosine learning-rate annealing, set the environment embedding dimension to 32, fix the low-frequency ratio to 0.40, use RK4 with step size 0.2, adopt a learning rate of 0.0005 and weight decay of 0.005 across datasets, and use 4 experts for Twitter and the relatively simpler tabular benchmarks (House, ONP, and Appliance) and 8 experts for Rotated MNIST, YearBook, Cyclone, and Shuttle. The remaining dataset-specific architectural and dimensional settings are as follows:

1. **Rotated MNIST.** We use 35 training domains, temporal segments of length 20, and a 15-step target horizon. The shared backbone is a 3-layer CNN with channel widths $[32, 32, 64]$, where each convolution is followed by ReLU and max pooling, and a dropout layer is applied after flattening. We train for 200 epochs with batch size 1. The latent dimension is 64, the hidden dimension is 256, the number of experts is 8, the spectral bin number is 5, and the Koopman rollout length is 15.

2. **Twitter.** We use 35 training domains, segment length 20, and a 15-step target horizon. The backbone first embeds the categorical state feature with a 10-dimensional embedding and then applies a 3-layer MLP with hidden widths 256 and 128 to produce a 64-dimensional representation. We train for 200 epochs with batch size 1. The latent dimension is 64, the hidden dimension is 128, the number of experts is 4, the spectral bin number is 4, and the Koopman rollout length is 10.

3. **YearBook.** We use 28 training domains, segment length 20, and a 12-step target horizon. The backbone is a 3-stage CNN with batch normalization and channel widths $[64, 128, 256]$, followed by a linear projection to 1024 dimensions and dropout. We train for 170 epochs with batch size 1. The latent dimension is 128, the hidden dimension is 256, the number of experts is 8, the spectral bin number is 5, and the Koopman rollout length is 15.

4. **Cyclone.** We use 50 training domains, segment length 20, and a 22-step target horizon. The backbone is a 4-layer CNN with channel widths $[32, 32, 64, 64]$, where each block uses ReLU and max pooling, followed by flattening and dropout. We train for 100 epochs

with batch size 1. The latent dimension is 128, the hidden dimension is 384, the number of experts is 8, the spectral bin number is 6, and the Koopman rollout length is 15.

5. **House.** We use 28 training domains, segment length 20, and a 12-step target horizon. For this tabular dataset, the backbone is the identity mapping and the predictor operates directly on the 50-dimensional input features. We train for 100 epochs with batch size 1. The latent dimension is 64, the hidden dimension is 256, the number of experts is 4, the spectral bin number is 5, and the Koopman rollout length is 6.
6. **ONP.** We use 5 training domains, temporal segments of length 4, and a 1-step target horizon. The model uses the raw 58-dimensional tabular features as input without an additional backbone. We train for 700 epochs with batch size 2. The latent dimension is 64, the hidden dimension is 256, the number of experts is 4, the spectral bin number is 5, and the Koopman rollout length is 3.
7. **Shuttle.** We use 7 training domains, temporal segments of length 4, and a 1-step target horizon. The model takes the 9-dimensional tabular features directly as input. We train for 350 epochs with batch size 2. The latent dimension is 64, the hidden dimension is 256, the number of experts is 8, the spectral bin number is 5, and the Koopman rollout length is 5.
8. **Appliance.** We use 8 training domains, temporal segments of length 4, and a 1-step target horizon. The model directly uses the 37-dimensional tabular features. We train for 500 epochs with batch size 2. The latent dimension is 64, the hidden dimension is 256, the number of experts is 4, the spectral bin number is 5, and the Koopman rollout length is 3.