LEARNING MULTIMODAL PREDICTIVE REGIONS WITH DIFFUSION: CONFORMAL PREDICTION FOR MULTIVARIATE TIME SERIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Multivariate time series forecasting informs decision making in domains such as traffic and energy systems. In these domains, effective decisions require understanding the uncertainty in future outcomes and the dependencies between them. To quantify this uncertainty, conformal prediction offers a model-agnostic way to calibrate multivariate prediction regions. However, it typically assumes exchangeability, which is generally violated by temporally dependent data. In addition, many commonly used conformal prediction methods relies on connected regions, which is inefficient for systems like traffic or energy models when the predictive distribution is multimodal or non-connected.

We address both issues using a conditional diffusion model within a conformal framework. Rather than running the reverse diffusion process, we use denoising errors to assess how well a possible future outcome y fits the learned conditional distribution. These errors are averaged into a scalar nonconformity score and calibrated using split conformal prediction. For finite samples from strictly stationary β-mixing time series, we establish bounds on marginal and conditional coverage, with a small penalty for temporal dependence. Experiments on non-Gaussian multimodal VAR processes and the Traffic and Solar datasets show competitive coverage/volume tradeoffs, with at least 7% smaller prediction regions in strongly multimodal settings.

1 INTRODUCTION

Traffic, energy, and environmental forecasting depend on point predictions, which do not quantify predictive uncertainty (Gneiting & Katzfuss, 2014). For example, both research efforts and industry practices in electric load forecasting have focused primarily on point load forecasting (Hong & Fan, 2016), while short-term traffic flow forecasting has been evaluated by point-prediction metrics like MAE and RMSE in Medina-Salgado et al. (2022). This limitation motivates a shift from point forecasts towards joint uncertainty quantification for multivariate outcomes, or conformal prediction (CP) (Shafer & Vovk, 2008; Vovk et al., 2005). It aims to construct regions with finite-sample coverage guarantees under suitable assumptions (Shafer & Vovk, 2008; Vovk et al., 2005).

However, such predictive distributions can be non-Gaussian and multimodal (Pinson, 2012; Tumu et al., 2024; Xiong et al., 2026). To address these issues, prior work has attempted to represent these predictive regions using standard shapes. Doing so either does not describe the relationship between dependent variables, such as with coordinate-wise intervals (Xu et al., 2024; Fan & Sesia, 2025), or overly covers low-confidence regions between modes of the distribution, as is the case with ellipsoidal regions (Messoudi et al., 2022). Improving on point forecasts therefore requires a method that can better represent joint non-Gaussian and multimodal predictive distributions.

Moreover, the standard validity guarantee for conformal prediction relies on exchangeability, which is violated by temporally dependent time series data (Lei et al., 2018). Existing work has developed conformal methods for temporally dependent data (Chernozhukov et al., 2021; Xu & Xie, 2021a; 2023; Oliveira et al., 2024) and flexible multivariate regions (Fang et al., 2025) separately. However, constructing flexible prediction regions for multivariate distributions while accounting for temporal dependence remains challenging.

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.

Promising work has now emerged using a different means to build multivariate, distributional geometry: score-based generative machine learning. Notably, diffusion models that learn to predict noise over time at different noise levels (Ho et al., 2020) can estimate score fields associated with a forward stochastic differential equation (SDE), as in conformal prediction (Song et al., 2020). This has led recent work to consider the use of diffusion as a prior for adaptive conformal inference under distribution shift (Jiang, 2026).

We propose to use diffusion directly to define a nonconformity score for temporally dependent multivariate responses. To that end, we introduce DISCO, a fully **DI**ffu**S**ion-based **CO**nformal predictor that creates non-convex or disconnected prediction regions defined as sublevel sets of the learned score. For strictly stationary β-mixing time series, DISCO admits finite-sample coverage guarantees that explicitly account for temporal dependence.

In addition to DISCO, our main contributions are:

- A novel nonconformity score based on denoising errors across multiple noise levels from DISCO, with direct and residual-based versions. (Section 3)
- Finite-sample marginal and localized conditional coverage guarantees for strictly stationary β-mixing sequences that account for sample size and temporal dependence. (Section 4)
- Extensive numerical and ablation studies of DISCO on synthetic VAR processes and the real-world Traffic and Solar datasets. (Section 5)

2 Related work

Conformal prediction for dependent sequences. Conformal prediction converts an arbitrary predictive model and nonconformity score into prediction sets with finite-sample marginal coverage under exchangeability (Lei et al., 2018). Exact feature-conditional coverage that is distribution free is generally impossible without additional assumptions (Barber et al., 2021), motivating guarantees tailored to sequential structure. Block-permutation methods preserve portions of the temporal dependence when computing conformal ranks (Chernozhukov et al., 2018). Distributional conformal prediction uses estimated percentile ranks to calibrate predictions, giving approximately correct coverage under suitable estimation and dependence conditions (Chernozhukov et al., 2021). Other approaches use bootstrap residuals under mixing assumptions (EnbPI) (Xu & Xie, 2021a), predict future conditional quantiles of nonconformity scores (SPCI) (Xu & Xie, 2023), adapt the target miscoverage level online (Gibbs & Candès, 2021), or apply weighted quantiles to reduce sensitivity to distributional drift (Barber et al., 2023). Most directly relevant to our analysis, Oliveira et al. (2024) shows that ordinary split conformal prediction remains approximately valid for broad classes of non-exchangeable data, with an explicit dependence penalty. We follow this route: we use chronological calibration and a fixed common-randomness bank to define a single measurable score. This enables a finite-sample marginal coverage bound for stationary β-mixing multivariate time series.

Multivariate and multimodal prediction regions. Scalar intervals do not directly capture dependence among simultaneous responses. MultiDimSPCI addresses this issue by using covariance-adaptive ellipsoidal regions and establishes bounds on the conditional coverage gap (Xu et al., 2024). Other approaches build conformal prediction regions by estimating the most likely outcome (Izbicki et al., 2022), or by optimizing structured templates for multimodal regions (Tumu et al., 2024). The closest generative baseline is FlowCP, which learns an invertible transformation conditioned on the input context and transports a Gaussian source ball into the response space (Lee et al., 2026). Although this is substantially more flexible than an ellipsoid, such a transported ball remains connected. In comparison, our prediction region is a sublevel set of a learned score-SDE nonconformity function, allowing flexible shapes and multiple disconnected regions. This distinction is central to the ring-mixture experiments, where efficient regions must exclude the low-density space between separated modes.

Score-based generative models for uncertainty quantification. Diffusion models learn denoising or noise-prediction functions across a sequence of perturbation levels (Ho et al., 2020). The score-SDE framework unifies these models through continuous-time forward and reverse stochastic processes (Song et al., 2020). Recent DiffConf work also repurposes diffusion scores for conformal

inference, focusing on online recalibration under temporal and covariate shift (Jiang, 2026). Our objective is complementary: we use denoising errors across noise levels to compute the nonconformity score for multivariate time-series responses, without requiring reverse-SDE sampling or an invertible transport. The residual formulation also separates the deterministic forecast structure from the geometry of the remaining uncertainty. Our experiments therefore evaluate on three complementary regimes: Gaussian innovations, where an ellipsoid is well specified; separated Gaussian mixtures, which test multimodal geometry; and the Traffic and Solar datasets, which test efficiency as the response dimension increases.

3 Problem Setup and Conformal Prediction

We consider a sequence of multivariate time series $\{(X_i, Y_i)\}_{i=1}^n$, where $X_i \in \mathbb{R}^{d_x}$ represents d_x-dimensional covariate variables and $Y_i \in \mathbb{R}^{d_y}$ represents d_y-dimensional response variables. Our goal is to construct a prediction set $\hat{C}(X_{n+1})$ given a new covariate variable X_{n+1} at time $n+1$ and significant level $\alpha \in (0,1)$. We aim to construct prediction sets that satisfy marginal coverage:

$$\mathbb{P}(Y_{n+1} \in \hat{C}(X_{n+1})) \geq 1 - \alpha \tag{1}$$

and localized conditional coverage given X_{n+1} in some neighborhood A:

$$\mathbb{P}(Y_{n+1} \in \hat{C}(X_{n+1}; A) \mid X_{n+1} \in A) \geq 1 - \alpha \tag{2}$$

A sequence of random elements $\{Z_t\}_{t=-\infty}^{\infty}$ taking values in a measurable space Z is stationary if its finite-dimensional distributions are time-invariant. That is, for any $t \in \mathbb{Z}$ and $m, k \in \mathbb{N}$,

$$Z_{t:(t+m)} = (Z_t, \ldots, Z_{t+m}) \stackrel{d}{=} (Z_{t+k}, \ldots, Z_{t+m+k}) = Z_{(t+k):(t+m+k)}.$$

Furthermore, for a stationary stochastic process $\{Z_t\}_{t=-\infty}^{\infty}$ and index $a \in \mathbb{N}$, the β-mixing coefficient of the process at a is defined as

$$\beta(a) = \left\| \mathbb{P}_{-\infty:0, a:\infty} - \mathbb{P}_{-\infty:0} \otimes \mathbb{P}_{a:\infty} \right\|_{TV}, \tag{3}$$

where $\|\cdot\|_{TV}$ denotes the total variation norm, and $P_{-\infty:0,a:\infty}$ is the joint distribution of the blocks $(Z_{-\infty:0}, Z_{a:\infty})$. The coefficient $\beta(a)$ measures the TV distance between the two multivariate distributions. The process is said to be β-mixing if $\beta(a) \to 0$ when $a \to \infty$.

3.1 Diffusion Processes and Score Functions

Let $X \in \mathbb{R}^{d_x}$ be the forecasting context and $Y_0 \in \mathbb{R}^{d_y}$ be the response. Following Song et al. (2020), we consider the forward diffusion

$$\mathrm{d}Y_t = f(Y_t, t)\,\mathrm{d}t + g(t)\,\mathrm{d}W_t, \qquad s_t(y \mid x) = \nabla_y \log p_t(y \mid x), \tag{4}$$

where $t \in [0, T]$ is the diffusion time, W_t is the Brownian motion independent of (X, Y_0), and $p_t(\cdot \mid x)$ is the conditional density of Y_t. The linear diffusion considered here admit the fixed-time representation $Y_t = m_t Y_0 + \sigma_t \varepsilon$, with $\varepsilon \sim \mathcal{N}(0, I_{d_y})$ independent of (X, Y_0). For the variance-preserving (VP) diffusion underlying DDPM (Ho et al., 2020), $f(y,t) = -g^2(t)y/2 = -\beta(t)y/2$, giving $m_t = e^{-\int_0^t \beta(u)\mathrm{d}u/2}$,$\sigma_t^2 = 1 - e^{-\int_0^t \beta(u)\mathrm{d}u}$. Constant $\beta(t)$ gives an Ornstein–Uhlenbeck (OU) process (Santos & Lin, 2023). For the variance-exploding (VE) diffusion associated with SMLD (Song & Ermon, 2019), $f = 0$, $m_t = 1$, and $\sigma_t^2 = \int_0^t g^2(u)\mathrm{d}u$, where σ_t^2 denotes the variance added to the clean response. Thus, VP attenuates the signal, whereas VE preserves its transition mean.

For $\sigma_t > 0$, the Gaussian transition score is $\nabla_y \log p_{t|0}(y \mid y_0) = -(y - m_t y_0)/\sigma_t^2$. Its conditional expectation gives the marginal score and the noise-prediction identity

$$s_t(y \mid x) = -\frac{\mathbb{E}[\varepsilon \mid Y_t = y, X = x]}{\sigma_t}, \qquad \widehat{s}_\theta(y, t \mid x) = -\frac{\widehat{\varepsilon}_\theta(y, t, x)}{\sigma_t}. \tag{5}$$

Under standard regularity conditions, $Z_\tau = Y_{T-\tau}$ satisfies the reverse SDE

$$\mathrm{d}Z_\tau = \left[-f(Z_\tau, T-\tau) + g^2(T-\tau) s_{T-\tau}(Z_\tau \mid x)\right]\mathrm{d}\tau + g(T-\tau)\mathrm{d}\overline{W}_\tau, \tag{6}$$

where τ increases from 0 to T, $Z_0 \sim p_T(\cdot \mid x)$, and $\overline{W}_\tau$ is the Brownian motion in reverse time. We use the learned denoiser to assess candidates directly, without simulating this reverse process.

3.2 Conformal Prediction for Time Series using Score SDEs

Training and common random numbers. We split the ordered pairs $\{(X_i, Y_i)\}_{i=1}^n$ at n_0 into training and calibration blocks, with $n_{\text{cal}} = n - n_0$. We then fit $\widehat{\varepsilon}_\theta$ only on the training block, using contexts available at each forecast origin. Next, we fix $0 < \delta < T$ and an even integer $R \geq 2$ without using calibration responses, and assume $\lambda(t) = \log(m_t^2/\sigma_t^2)$ is continuous and strictly decreasing on $[\delta, T]$. After training, we generate a single common-random-number (CRN) bank $w = \{t_k, \varepsilon_{k,r}\}_{k \leq K, r \leq R}$ independently of the data and training randomness, as specified in Algorithm 1. The times are stratified uniformly in log-SNR, and the Gaussian perturbations form antithetic pairs. For VP, we obtain t_k by inverting $B(t_k) = \log(1 + e^{-\lambda_k})$, and for VE, we invert $\sigma_{t_k}^2 = e^{-\lambda_k}$. This includes constant-rate OU without imposing a special rate on general DDPM schedules.

Nonconformity and calibration. We reserve s_t for the diffusion score and define the scalar nonconformity score

$$s_w(x, y) = \frac{1}{|\mathcal{T}|R} \sum_{k=1}^{|\mathcal{T}|} \sum_{r=1}^{R} \|\varepsilon_{k,r} - \widehat{\varepsilon}_\theta(m_{t_k} y + \sigma_{t_k} \varepsilon_{k,r}, t_k, x)\|_2^2 . \tag{7}$$

By reusing the same bank for every calibration pair and test candidate and disabling inference-time randomness, we ensure that s_w is deterministic given the fitted network and w. Smaller values indicate greater denoising consistency, although this ordering need not coincide with conditional density ordering. Let $s_{(1)} \leq \cdots \leq s_{(n_{\text{cal}})}$ be the ordered calibration scores $s_i = s_w(X_i, Y_i)$, and set

$$\widehat{q}_{1-\alpha} = s_{\lceil (n_{\text{cal}}+1)(1-\alpha) \rceil}, \qquad s_{(n_{\text{cal}}+1)} := +\infty. \tag{8}$$

The prediction region is

$$\widehat{\mathcal{C}}_{1-\alpha}(x) = \{y \in \mathbb{R}^{d_y} : s_w(x, y) \leq \widehat{q}_{1-\alpha}\}. \tag{9}$$

Conditional exchangeability of calibration and test pairs given the fitted model yields the usual marginal coverage bound of at least $1 - \alpha$. For dependent time series, chronological splitting and shared randomization alone do not ensure this guarantee. Additional dependence assumptions and a corresponding coverage analysis are required.

Algorithm 1 Conformal prediction via score-based SDE denoising

Require: Time series data $\mathcal{D}$; split n_0; test context X_{n+1}; $\alpha \in (0, 1)$; kernel (m_t, σ_t); $0 < \delta < T$; invertible decreasing log-SNR λ; even number $R \geq 2$.

1: Train $\widehat{\varepsilon}_\theta$ on the first n_0 pairs; set $n_{\text{cal}} \leftarrow n - n_0$.
2: **for** $k = 1, \ldots, |\mathcal{T}|$ **do**
3: Draw $U_k \sim \mathcal{U}(0, 1)$ independently; set $v_k \leftarrow (k - 1 + U_k)/|\mathcal{T}|$.
4: Set $\lambda_k \leftarrow (1 - v_k)\lambda(\delta) + v_k \lambda(T)$ and $t_k \leftarrow \lambda^{-1}(\lambda_k)$.
5: Draw independent $\varepsilon_{k,r} \sim \mathcal{N}(0, I_{d_y})$, $r \leq R/2$; set $\varepsilon_{k,r+R/2} \leftarrow -\varepsilon_{k,r}$.
6: **end for**
7: Freeze $w = \{t_k, \varepsilon_{k,r}\}_{k \leq |\mathcal{T}|, r \leq R}$, generated independently of data and training.
8: Compute $s_i \leftarrow s_w(X_i, Y_i)$ for $i = n_0 + 1, \ldots, n$ using equation 7.
9: Set $\widehat{q}_{1-\alpha}$ by equation 8, including the $+\infty$ convention.
10: **return** $\widehat{\mathcal{C}}_{1-\alpha}(X_{n+1})$ in equation 9, using the same w for all candidates.

Figure 1 compares the geometry of the prediction regions in the multimodal VAR(1) example. Our methods capture the separated modes more closely, whereas the competing methods shown produce broader disk- or ring-shaped regions.

4 Theory

4.1 Finite-budget approximation of the empirical nonconformity score

The approximation error of the empirical nonconformity score s used in Algorithm 1 has two components: Monte Carlo error conditional on the trained denoiser, and estimation error relative to the oracle denoiser.

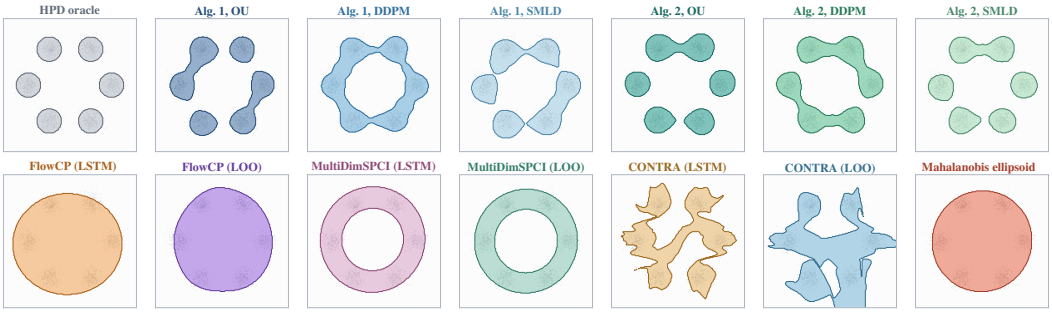


Figure 1: Comparison of 95% prediction regions for the VAR(1) Gaussian-mixture example with 8.3σ separation and $n = 8000$. From left to right and top to bottom: the oracle HPD region, our Algorithm 1, our residual-based Algorithm 2, FlowCP with the LSTM predictor and encoder, FlowCP with the LOO-bootstrap predictor and encoder, MultiDimSPCI with the LSTM predictor, MultiDimSPCI with the LOO-bootstrap predictor, and the Mahalanobis ellipsoid.

For $\varepsilon \sim \mathcal{N}(0, I_{d_y})$, we define the denoising loss as $\ell_\theta(x, y; v, \varepsilon) := \|\varepsilon - \varepsilon_\theta(m_{t(v)}y + \sigma_{t(v)}\varepsilon, t(v), x)\|_2^2$. Conditional on the training sample, we then define the estimated population score $s_{\widehat{\theta}}(x, y) = \mathbb{E}_{v,\varepsilon}\left[\ell_{\widehat{\theta}}(x, y; v, \varepsilon)\right]$, where $v \sim \mathcal{U}(0, 1)$. Now we impose the finite-budget regularity conditions.

Assumption 1 (Finite-budget regularity)**.** *For any fixed (x, y), conditional on the training sample $\mathcal{D}_{\text{train}}$,*

$$\sup_{v\in[0,1]} \mathit{Var}_\varepsilon[\ell_{\hat{\theta}}(x, y; v, \varepsilon) + \ell_{\hat{\theta}}(x, y; v, -\varepsilon)] \le C(x, y) < \infty$$

The finite constants may depend measurably on the estimated network. Furthermore, assume that the population loss along the log-SNR scale $h_{\widehat{\theta},x,y}(v) = \mathbb{E}_\varepsilon\left[\ell_{\widehat{\theta}}(x, y; v, \varepsilon)\right]$ is $L(x, y)$-Lipschitz on $[0, 1]$.

Theorem 4.1 (Finite-budget score approximation)**.** *Under Assumption 1, the expectation and probability are over the randomized bank w, conditional on the training sample $\mathcal{D}_{\text{train}}$,*

$$\mathbb{E}_w\left[\left\{s_w(x, y) - s_{\widehat{\theta}}(x, y)\right\}^2\right] \le \frac{C(x, y)}{2|\mathcal{T}|R} + \frac{L^2(x, y)}{12|\mathcal{T}|^3}. \tag{10}$$

Moreover, for any $\delta_{\text{mc}} \in (0, 1)$, with probability at least $1 - \delta_{\text{mc}}$,

$$\left|s_w(x, y) - s_{\widehat{\theta}}(x, y)\right| \le \left\{\frac{1}{\delta_{\text{mc}}}\left(\frac{C(x, y)}{2|\mathcal{T}|R} + \frac{L^2(x, y)}{12|\mathcal{T}|^3}\right)\right\}^{1/2}. \tag{11}$$

Now, we will discuss the gap between the approximated score and the oracle population score under β-mixing processes. First, we define the corresponding oracle population score based on the oracle denoising loss ℓ_0, $s_0(x, y) = \mathbb{E}_{v,\varepsilon}\left[\ell_0(x, y; v, \varepsilon)\right]$, where $\ell_0(x, y; v, \varepsilon) = \|\varepsilon - \varepsilon_0(m_{t(v)}y + \sigma_{t(v)}\varepsilon, t(v), x)\|_2^2$, where $\varepsilon_0(z, t, x) = \mathbb{E}[\varepsilon \mid m_t Y + \sigma_t \varepsilon = z, X = x]$. Unlike the finite-budget approximation above, comparison with $s_0(x, y)$ requires a learning condition on the estimated denoiser. Instead of imposing uniform consistency over the entire input space, we only require an integrated prediction error along the perturbation distribution relevant to the fixed candidate (x, y). Define the denoiser approximation

$$d^2(x, y, \widehat{\theta}) = \mathbb{E}_{v,\varepsilon}\left\|\varepsilon_{\widehat{\theta}}(m_{t(v)}y + \sigma_{t(v)}\varepsilon, t(v), x) - \varepsilon_0(m_{t(v)}y + \sigma_{t(v)}\varepsilon, t(v), x)\right\|_2^2$$

and the ball on (x, y)

$$B_0(x, y) = \left\{\mathbb{E}_{v,\varepsilon}\left[\left\|\varepsilon - \varepsilon_0\left(m_{t(v)}y + \sigma_{t(v)}\varepsilon, t(v), x\right)\right\|_2^2\right]\right\}^{1/2}.$$

For $1 \le a \le \lfloor n_0/2 \rfloor$, put $m = \lfloor n_0/(2a) \rfloor$ and $B_j^{(q)} = (Z_{(2j-2+q)a+1}, \ldots, Z_{(2j-1+q)a})$, $q \in \{0, 1\}$, $1 \le j \le m$. Within each parity q, let $\mathbf{B}^{(q,*)}$ have independent blocks with the original block marginals; observations within each block retain their dependence.

Assumption 2. *For the actual estimator $\hat{\theta}$, $\delta_0 \in (0,1)$ and a deterministic radius $r = r_{m,a}(\delta_0; x, y)$, there are measurable block events $\mathcal{E}_q$, indexed by a nonempty $Q \subseteq \{0,1\}$, such that*

$$\{d(x, y, \widehat{\theta}) > r\} \subseteq \bigcup_{q \in Q} \{\mathbf{B}^{(q)} \in \mathcal{E}_q\}, \qquad \sum_{q \in Q} \mathbb{P}(\mathbf{B}^{(q,*)} \in \mathcal{E}_q) \leq \delta_0. \tag{12}$$

Lemma 4.1 (Learning bound under β-mixing process)**.** *Under Assumption 2,*

$$\mathbb{P}\{d(x, y, \widehat{\theta}) > r_{m,a}(\delta_0; x, y)\} \leq \delta_0 + |Q|(m-1)\beta(a), \tag{13}$$

where $|Q|$ denotes the number of elements in the set Q defined in the Assumption 2.

Corollary 4.2 (Score approximation)**.** *Suppose Assumption 1 holds almost surely and Assumption 2 holds for the estimator $\hat{\theta}$. For $\delta_{\mathrm{mc}}, \delta_0 \in (0,1)$ and $r = r_{m,a}(\delta_0; x, y)$, with joint probability at least $1 - \delta_{\mathrm{mc}} - \delta_0 - |Q|(m-1)\beta(a)$,*

$$|s_w(x,y) - s_0(x,y)| \leq \underbrace{\left\{ \frac{1}{\delta_{\mathrm{mc}}} \left(\frac{C(x,y)}{2|\mathcal{T}|R} + \frac{L^2(x,y)}{12|\mathcal{T}|^3} \right) \right\}^{1/2}}_{\textit{Monte Carlo error}} + \underbrace{2B_0(x,y)r + r^2}_{\textit{plug-in error}}. \tag{14}$$

Remark 4.1 (Geometric mixing)**.** *If $\beta(a) \leq Ce^{-ca}$, choosing $a = \lceil \kappa \log n_0 \rceil$ with $\kappa > 1/c$ gives $m \asymp n_0 / \log n_0$ and $(m-1)\beta(a) \leq Cn_0^{1-c\kappa}/(2a) = o(1)$. The rate $r_{m,a}$ still requires an independent-block learning result.*

4.2 Marginal Coverage

In this section, we will establish the finite-sample approximate marginal coverage for the prediction sets by our method for stationary β-mixing processes. Under a CRN bank w, define the probability law $\mathbb{P}_w(\cdot) = \mathbb{P}(\cdot \mid W = w)$ where W is a random variable independent of the data $\mathcal{D}_{\mathrm{train}}$ and sampled before calibration.

Theorem 4.3 (Marginal coverage guarantee under stationary β-mixing processes)**.** *Let $Z_i = (X_i, Y_i) \in \mathbb{R}^{d_x} \times \mathbb{R}^{d_y}$, $i \in \mathbb{Z}$, be a strictly stationary β-mixing process with mixing coefficients $\beta(a)$ defined in equation 3. Let $I_{train} = \{1, \ldots, n_0\}$, $I_{\mathrm{cal}} = \{n_0 + 1, \ldots, n\}$, and $n_{\mathrm{cal}} = n - n_0$. Then given $\alpha, \delta_{\mathrm{cal}} \in (0,1)$, for any CRN bank w, we have*

$$\mathbb{P}_w(Y_{n+1} \in \hat{C}^w_{1-\alpha}(X_{n+1})) \geq 1 - \alpha - \varepsilon_{\mathrm{cal}} - \delta_{\mathrm{cal}} - \beta(n_{\mathrm{cal}} + 1),$$

where

$$\begin{aligned} \varepsilon_{\mathrm{cal}} = \inf_{(a,m,r) \in \mathcal{F}_{\mathrm{cal}}} \Bigg\{ & \widetilde{\sigma}(a) \sqrt{\frac{4}{n_{\mathrm{cal}} - r + 1} \log\left(\frac{4}{\delta_{\mathrm{cal}} - 4(m-1)\beta(a) - \beta(r)} \right)} \\ & + \frac{1}{3m} \log\left(\frac{4}{\delta_{\mathrm{cal}} - 4(m-1)\beta(a) - \beta(r)} \right) + \frac{r-1}{n_{\mathrm{cal}}} \Bigg\} \end{aligned} \tag{15}$$

with non-empty

$$\mathcal{F}_{\mathrm{cal}} = \left\{ (a,m,r) \in \mathbb{N}^3_{>0} : 2ma = n_{\mathrm{cal}} - r + 1,\ \delta_{\mathrm{cal}} > 4(m-1)\beta(a) + \beta(r) \right\} \tag{16}$$

and

$$\widetilde{\sigma}(a) = \sqrt{\frac{1}{4} + \frac{2}{a} \sum_{j=1}^{a-1} (a-j)\beta(j)}. \tag{17}$$

Furthermore, we have

$$\mathbb{P}[Y_{n+1} \in \widehat{C}_{1-\alpha}(X_{n+1})] \geq 1 - \alpha - \eta,$$

where $\eta = \varepsilon_{\mathrm{cal}} + \delta_{\mathrm{cal}} + \beta(n_{\mathrm{cal}} + 1)$.

4.3 CONDITIONAL GUARANTEES

Barber et al. (2021) prove that conditional coverage is not exactly attainable, even for i.i.d. data, but can be achieved for some specific sets with finite VC dimension. Our work is to provide localized conditional guarantees for score-based conformal prediction under multivariate time series.

For continuous covariates, arbitrarily small neighborhoods may have vanishing probability mass. Therefore, rather than considering all Euclidean balls, we first define the family of all Euclidean balls $\mathcal{A} := \{B(x,\rho) : x \in \mathbb{R}^{d_x},\ \rho > 0\}, \qquad B(x,\rho) := \{u \in \mathbb{R}^{d_x} : \|u - x\|_2 \le \rho\}$. The VC dimension for the family is at most $d_x + 1$. Therefore, define $\mathcal{A}_\gamma := \{A \in \mathcal{A} : \mathbb{P}(X^* \in A) \ge \gamma\}$. For any CRN bank w, each $A \in \mathcal{A}_\gamma$, define

$$\hat{C}^w_{1-\alpha}(X; A) := \{Y : s_w(X,Y) \le q^w_{1-\alpha}(A)\}$$

where $q^w_{1-\alpha}(A)$ is $\lceil (n_{\text{cal}}(A)+1)(1-\alpha) \rceil$-th order statistic of $\{s_w(Z_i) : i \in I_{\text{cal}}(A)\}$ with $I_{\text{cal}}(A) = \{i \in I_{\text{cal}} : X_i \in A\}$ and $n_{\text{cal}}(A) = |I_{\text{cal}}(A)|$. If $\lceil (n_{\text{cal}}(A)+1)(1-\alpha) \rceil > n_{\text{cal}}(A)$, set $q^w_{1-\alpha}(A) = +\infty$. This convention also applies when $n_{\text{cal}}(A) = 0$.

Theorem 4.4 (Conditional coverage guarantee under stationary β-mixing processes)**.** *Let $Z_i = (X_i, Y_i)$ be a strictly stationary β-mixing process. Fix $\alpha, \delta_{\text{cal}}, \gamma \in (0,1)$. For each fixed $A \in \mathcal{A}_\gamma$ and each fixed CRN bank w, we have*

$$\mathbb{P}_w(Y_{n+1} \in \hat{C}^w_{1-\alpha}(X_{n+1}; A) \mid X_{n+1} \in A) \ge 1 - \alpha - \varepsilon_{\text{cal}} - \delta_{\text{cal}}/\gamma - \beta(n_{\text{cal}}+1)/\gamma,$$

where

$$\varepsilon_{\text{cal}} = \frac{1}{\gamma} \inf_{(a,m,r) \in \mathcal{G}_{\text{cal}}} \left\{ 4\sqrt{\frac{\log(2(m+1)^{d_x+1})}{m}} + \frac{4(r-1)}{n_{\text{cal}}} + \sqrt{\frac{2}{m} \log\left(\frac{16}{\delta_{\text{cal}} - 16(m-1)\beta(a) - \beta(r)}\right)} \right\} < 1 \tag{18}$$

with non-empty

$$\mathcal{G}_{\text{cal}} = \{(a,m,r) \in \mathbb{N}^3_{>0} : 2ma = n_{\text{cal}} - r + 1, \delta_{\text{cal}} > 16(m-1)\beta(a) + \beta(r)\}. \tag{19}$$

5 REAL DATA EXPERIMENTS

We evaluate the conformal prediction for multivariate time-series prediction tasks using two real-world datasets. We consider our methods: Algorithm 1 using the conditional diffusion models within the conformal framework, and the residual variant, i.e. Algorithm 2. We compare these methods under different SDEs, i.e. OU, DDPM and SMLD with other conformal baselines. All experiments are run on 1 node of 8 NVIDIA H100 GPUs, each with 80GB memory.

Baselines. We compare our algorithms with four baseline families. First, we consider two variants of FlowCP Lee et al. (2026) using the authors' official implementation. Next, we consider Multi-DimSPCI Xu et al. (2024), CONTRA Fang et al. (2025) with the same LSTM predictor and LOO bootstrap. We also include a global conformal Mahalanobis ellipsoid. Further details on the baseline implementations and hyperparameter settings are provided in Appendix B.1.

Datasets. We evaluate Algorithms 1 and 2 and the baselines on two real-world time series datasets: Traffic and Solar datasets. For each dataset, we consider $d_y \in \{2, 4\}$ and construct five reproducible location subsets. Within each subset, we sample d_y distinct locations without replacement using a fixed random seed. The one-step prediction uses a window of 50 based on the training observations. For stationary assumptions and no data leakage, we do the transformation for each dataset based on the mean and standard deviation only from the training dataset. Additional dataset details are provided in Appendix B.3.

Evaluation metrics. We use two evaluation metrics: empirical coverage and the average prediction volume. The empirical coverage under a target confidence level α is defined by

$$CVR_{1-\alpha} = \frac{1}{n_{\text{test}}} \sum_{i \in I_{\text{test}}} 1\left\{Y_i \in \widehat{C}_{1-\alpha}(X_i)\right\}. \tag{20}$$

The average prediction region volume is computed by averaging the $\hat{C}$ across the five location folds. All volumes are measured using the common importance-sampling procedure described in Appendix B.4.

Results. Table 1 summarizes the real-data results. Algorithm 1 produces the smallest average prediction-region volume across both datasets and response dimensions, although its empirical coverage is often below the nominal level. Among three different SDEs, SMLD often yields the smallest regions for Algorithm 1. The residual-based Algorithm 2 produces larger regions but reduces coverage variability on Traffic and improves average coverage in most settings compared to Algorithm 1.

On Traffic, it achieves mean coverage at or above the nominal 95% and 90% levels while producing smaller regions than FlowCP and the Mahalanobis ellipsoid. Among the other baselines, only CONTRA with LOO bootstrap achieves the nominal 95% level for the Traffic dataset with $d_y = 4$, and its volume size is still larger than Algorithm 1. On Solar, however, the mean coverages by Algorithm 1 and 2 remain below the nominal 90% level. For the residual-based algorithm, no single diffusion backbone consistently yields the smallest volumes.

The empirical undercoverage is consistent with the finite-sample coverage penalties induced by temporal dependence in our theoretical lower bound. In addition, score approximation provides another source of finite-sample discrepancy. In particular, it illustrates that nominal conformal coverage need not be attained exactly in finite samples when exchangeability is violated in the time series data. It also suggests a tradeoff between prediction-region volume and empirical coverage. In particular, the smaller regions produced by Algorithm 1 tend to exhibit undercoverage in time series data settings, whereas the residual-based Algorithm 2 sacrifices some volume efficiency to achieve more coverage.

Ablation study. We conduct an ablation study to assess the contribution of the randomized score-construction components: stratification, antithetic Gaussian sampling, and CRN. More details about the ablation study can be found in Appendix C. Tables 7 and Table 8 report the average empirical coverage, prediction-region volume, and Monte Carlo Score SD of Algorithms 1 and 2 under the OU diffusion backbone on the real-world datasets and the 8.3σ multimodal VAR(1) process, respectively. We observe that removing either antithetic sampling or the CRN bank led to larger prediction region volumes. Algorithm 2 produces larger regions than Algorithm 1 on the real-world datasets and improves coverage on Traffic, but its coverage remains below the nominal 90% level on Solar. Its relative volume efficiency on synthetic data varies with sample size, as shown in Table 6.

6 Discussion and Conclusion

We introduced a new conformal prediction method for multivariate time series that uses diffusion denoising errors to define its nonconformity score. Coverage guarantees of our methods are provided by establishing non-asymptotic bounds on marginal and conditional coverage with a small penalty for temporal dependence. Experiments on multimodal VAR processes demonstrate that DISCO provides compact prediction regions with flexible shapes under multivariate time series. On the Traffic and Solar datasets, DISCO yields the smallest mean prediction-region volumes, with mean empirical coverage below nominal in several settings. The residual variant produces larger regions and exhibits lower coverage variability on Traffic, although its mean empirical coverage remains below 90% on Solar at that target. These results suggest a practical tradeoff between prediction-region volume and empirical coverage, motivating future studies on real-world time series with multivariate and multimodel response distributions.

One challenge of DISCO is stationarity and β-mixing conditions, which may not hold under seasonality or distribution shift in the non-stationary time series. Future work can investigate diffusion-based conformal prediction for nonstationary time series. Moreover, it remains difficult to obtain efficient prediction regions as the dimension of datasets grows. Future work can develop more efficient score approximations for higher-dimensional time-series responses.

Author Contributions

Censored for review

Table 1: Prediction regions on the Traffic and Solar datasets with a shared training-only per-location standardization. CVR (%) and volume are reported as mean ± one standard deviation over five location folds. **Bold** and underlining denote the smallest and second-smallest mean volumes, respectively, for each dataset, d_y, and α.

			$\alpha = 0.05$		$\alpha = 0.10$	
Dataset	d_y	Method	CVR (%)	Volume	CVR (%)	Volume
Traffic	2	Ours (Algorithm 1, OU)	$93.7_{\pm 5.5}$	$\underline{0.501_{\pm 0.335}}$	$87.9_{\pm 8.5}$	$\underline{0.292_{\pm 0.186}}$
		Ours (Algorithm 1, DDPM)	$91.9_{\pm 9.0}$	$0.526_{\pm 0.356}$	$86.1_{\pm 11.5}$	$0.299_{\pm 0.208}$
		Ours (Algorithm 1, SMLD)	$92.4_{\pm 7.9}$	$\mathbf{0.455_{\pm 0.272}}$	$86.8_{\pm 11.5}$	$\mathbf{0.283_{\pm 0.178}}$
		Ours (Algorithm 2, OU)	$96.8_{\pm 1.5}$	$0.865_{\pm 0.340}$	$92.7_{\pm 3.9}$	$0.529_{\pm 0.209}$
		Ours (Algorithm 2, DDPM)	$96.6_{\pm 1.8}$	$0.867_{\pm 0.324}$	$92.8_{\pm 3.2}$	$0.533_{\pm 0.207}$
		Ours (Algorithm 2, SMLD)	$97.0_{\pm 2.0}$	$0.897_{\pm 0.334}$	$93.0_{\pm 3.3}$	$0.542_{\pm 0.196}$
		FlowCP (LSTM + encoder)	$96.8_{\pm 2.6}$	$2.120_{\pm 1.125}$	$91.9_{\pm 3.7}$	$1.316_{\pm 0.845}$
		FlowCP (LOO bootstrap + encoder)	$95.7_{\pm 1.9}$	$1.734_{\pm 0.958}$	$90.9_{\pm 2.3}$	$1.175_{\pm 0.641}$
		MultiDimSPCI (LSTM)	$87.3_{\pm 3.6}$	$1.357_{\pm 1.237}$	$85.3_{\pm 5.6}$	$1.067_{\pm 1.028}$
		MultiDimSPCI (LOO bootstrap)	$90.3_{\pm 3.8}$	$0.574_{\pm 0.156}$	$88.3_{\pm 2.8}$	$0.396_{\pm 0.113}$
		CONTRA (LSTM)	$93.1_{\pm 3.6}$	$0.934_{\pm 0.767}$	$83.9_{\pm 8.4}$	$0.536_{\pm 0.425}$
		CONTRA (LOO bootstrap)	$94.4_{\pm 2.3}$	$0.539_{\pm 0.301}$	$89.0_{\pm 3.3}$	$0.326_{\pm 0.180}$
		Mahalanobis ellipsoid	$96.9_{\pm 1.9}$	$0.966_{\pm 0.420}$	$92.8_{\pm 2.6}$	$0.577_{\pm 0.221}$
	4	Ours (Algorithm 1, OU)	$93.2_{\pm 9.5}$	$\underline{0.524_{\pm 0.343}}$	$87.8_{\pm 10.9}$	$\underline{0.209_{\pm 0.119}}$
		Ours (Algorithm 1, DDPM)	$92.6_{\pm 10.2}$	$0.594_{\pm 0.301}$	$86.6_{\pm 15.1}$	$0.248_{\pm 0.126}$
		Ours (Algorithm 1, SMLD)	$95.2_{\pm 4.5}$	$\mathbf{0.493_{\pm 0.358}}$	$89.3_{\pm 7.6}$	$\mathbf{0.189_{\pm 0.129}}$
		Ours (Algorithm 2, OU)	$97.1_{\pm 1.8}$	$1.230_{\pm 0.732}$	$93.7_{\pm 2.3}$	$0.512_{\pm 0.299}$
		Ours (Algorithm 2, DDPM)	$96.9_{\pm 1.5}$	$1.275_{\pm 0.754}$	$93.3_{\pm 2.5}$	$0.509_{\pm 0.301}$
		Ours (Algorithm 2, SMLD)	$96.9_{\pm 1.7}$	$1.178_{\pm 0.681}$	$93.5_{\pm 2.4}$	$0.507_{\pm 0.288}$
		FlowCP (LSTM + encoder)	$96.0_{\pm 2.7}$	$7.874_{\pm 4.596}$	$91.2_{\pm 3.9}$	$4.097_{\pm 2.079}$
		FlowCP (LOO bootstrap + encoder)	$96.3_{\pm 1.1}$	$6.681_{\pm 5.285}$	$92.3_{\pm 2.0}$	$2.655_{\pm 1.089}$
		MultiDimSPCI (LSTM)	$84.7_{\pm 1.7}$	$6.505_{\pm 4.405}$	$82.9_{\pm 3.4}$	$4.220_{\pm 2.898}$
		MultiDimSPCI (LOO bootstrap)	$82.3_{\pm 5.1}$	$0.676_{\pm 0.319}$	$80.7_{\pm 4.8}$	$0.352_{\pm 0.161}$
		CONTRA (LSTM)	$91.7_{\pm 8.1}$	$2.768_{\pm 1.934}$	$82.8_{\pm 13.3}$	$1.108_{\pm 0.998}$
		CONTRA (LOO bootstrap)	$95.1_{\pm 1.0}$	$0.791_{\pm 0.353}$	$89.9_{\pm 1.4}$	$0.338_{\pm 0.144}$
		Mahalanobis ellipsoid	$97.0_{\pm 1.8}$	$1.438_{\pm 0.867}$	$93.1_{\pm 2.8}$	$0.602_{\pm 0.347}$
Solar	2	Ours (Algorithm 1, OU)	$94.6_{\pm 0.4}$	$\mathbf{0.295_{\pm 0.067}}$	$88.6_{\pm 1.4}$	$\underline{0.119_{\pm 0.017}}$
		Ours (Algorithm 1, DDPM)	$94.1_{\pm 0.5}$	$0.360_{\pm 0.134}$	$88.2_{\pm 1.4}$	$0.121_{\pm 0.033}$
		Ours (Algorithm 1, SMLD)	$95.0_{\pm 0.4}$	$\underline{0.314_{\pm 0.107}}$	$88.2_{\pm 1.5}$	$\mathbf{0.107_{\pm 0.022}}$
		Ours (Algorithm 2, OU)	$95.2_{\pm 0.9}$	$0.973_{\pm 0.145}$	$88.6_{\pm 1.5}$	$0.337_{\pm 0.048}$
		Ours (Algorithm 2, DDPM)	$94.7_{\pm 0.5}$	$1.054_{\pm 0.319}$	$88.0_{\pm 2.1}$	$0.346_{\pm 0.067}$
		Ours (Algorithm 2, SMLD)	$95.1_{\pm 0.6}$	$0.957_{\pm 0.131}$	$89.1_{\pm 1.7}$	$0.374_{\pm 0.082}$
		FlowCP (LSTM + encoder)	$95.3_{\pm 0.4}$	$1.432_{\pm 0.477}$	$89.9_{\pm 1.6}$	$0.724_{\pm 0.371}$
		FlowCP (LOO bootstrap + encoder)	$94.8_{\pm 1.2}$	$1.179_{\pm 0.338}$	$89.5_{\pm 1.7}$	$0.734_{\pm 0.258}$
		MultiDimSPCI (LSTM)	$95.6_{\pm 0.4}$	$0.867_{\pm 0.156}$	$92.2_{\pm 0.9}$	$0.541_{\pm 0.110}$
		MultiDimSPCI (LOO bootstrap)	$94.2_{\pm 0.6}$	$0.995_{\pm 0.216}$	$91.4_{\pm 0.5}$	$0.622_{\pm 0.142}$
		CONTRA (LSTM)	$93.2_{\pm 0.6}$	$1.430_{\pm 0.342}$	$86.1_{\pm 2.3}$	$0.626_{\pm 0.072}$
		CONTRA (LOO bootstrap)	$92.2_{\pm 1.2}$	$1.781_{\pm 1.599}$	$87.1_{\pm 1.9}$	$0.588_{\pm 0.184}$
		Mahalanobis ellipsoid	$95.6_{\pm 0.5}$	$0.976_{\pm 0.153}$	$90.1_{\pm 1.4}$	$0.419_{\pm 0.067}$
	4	Ours (Algorithm 1, OU)	$94.9_{\pm 0.9}$	$\underline{0.297_{\pm 0.084}}$	$88.1_{\pm 0.8}$	$\mathbf{0.039_{\pm 0.015}}$
		Ours (Algorithm 1, DDPM)	$95.1_{\pm 0.5}$	$0.610_{\pm 0.268}$	$87.3_{\pm 1.6}$	$0.058_{\pm 0.031}$
		Ours (Algorithm 1, SMLD)	$94.4_{\pm 0.6}$	$\mathbf{0.274_{\pm 0.099}}$	$87.6_{\pm 0.8}$	$\underline{0.041_{\pm 0.023}}$
		Ours (Algorithm 2, OU)	$95.6_{\pm 0.6}$	$1.129_{\pm 0.546}$	$88.4_{\pm 1.5}$	$0.155_{\pm 0.051}$
		Ours (Algorithm 2, DDPM)	$95.2_{\pm 0.3}$	$0.955_{\pm 0.379}$	$88.5_{\pm 1.0}$	$0.156_{\pm 0.056}$
		Ours (Algorithm 2, SMLD)	$95.5_{\pm 0.7}$	$1.237_{\pm 0.541}$	$88.9_{\pm 2.0}$	$0.193_{\pm 0.110}$
		FlowCP (LSTM + encoder)	$94.8_{\pm 0.7}$	$3.916_{\pm 3.434}$	$90.4_{\pm 0.9}$	$1.292_{\pm 1.321}$
		FlowCP (LOO bootstrap + encoder)	$93.8_{\pm 1.0}$	$2.096_{\pm 0.875}$	$89.3_{\pm 0.8}$	$1.143_{\pm 0.657}$
		MultiDimSPCI (LSTM)	$94.3_{\pm 1.2}$	$3.933_{\pm 1.146}$	$91.9_{\pm 1.1}$	$1.863_{\pm 0.625}$
		MultiDimSPCI (LOO bootstrap)	$94.3_{\pm 0.9}$	$4.253_{\pm 1.256}$	$92.4_{\pm 1.1}$	$2.002_{\pm 0.761}$
		CONTRA (LSTM)	$94.8_{\pm 1.6}$	$10.776_{\pm 5.706}$	$90.6_{\pm 2.6}$	$2.833_{\pm 1.350}$
		CONTRA (LOO bootstrap)	$95.1_{\pm 1.8}$	$13.328_{\pm 16.480}$	$89.6_{\pm 2.8}$	$3.876_{\pm 4.481}$
		Mahalanobis ellipsoid	$95.9_{\pm 0.6}$	$1.198_{\pm 0.468}$	$90.3_{\pm 1.6}$	$0.217_{\pm 0.077}$

ACKNOWLEDGMENTS

Censored for review

ETHICS STATEMENTS

Our work complies with the ICLR Code of Ethics. We do not have other ethical issues.

AI USE

Generative AI tools were not used in the core scientific process, including but not limited to idealization, writing the proofs, or designing experiments. It was used only for writing software used for running experiments. All code generated by LLMs had to be manually reviewed by both the user and another person on the team before being accepted into our codebase.

For the paper LLM assistance was used for cleaning up rough drafts of the paper. All edits to the paper were manually reviewed by a team of at least 2 people and a second pass with a human writer was done to ensure consistency in the paper. In many cases, we found the human review often overwrote the LLM-assisted cleanup entirely.

We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The source code for the method and experiments is available at `https://anonymous.4open.science/r/Disco-ts`. Details of the experimental setup, including hyperparameters and datasets are provided in the Experiments section and Appendix B.

REFERENCES

Rina Foygel Barber, Emmanuel J. Candès, Aaditya Ramdas, and Ryan J. Tibshirani. The limits of distribution-free conditional predictive inference. *Information and Inference: A Journal of the IMA*, 10(2):455–482, 2021. URL https://doi.org/10.1093/imaiai/iaaa017.

Rina Foygel Barber, Emmanuel J. Candès, Aaditya Ramdas, and Ryan J. Tibshirani. Conformal prediction beyond exchangeability. *The Annals of Statistics*, 51(2):816–845, 2023. URL https://doi.org/10.1214/23-AOS2276.

Victor Chernozhukov, Kaspar Wüthrich, and Yinchu Zhu. Exact and robust conformal inference methods for predictive machine learning with dependent data. In *Proceedings of the 31st Conference on Learning Theory*, volume 75 of *Proceedings of Machine Learning Research*, pp. 732–749. PMLR, 2018. URL https://proceedings.mlr.press/v75/chernozhukov18a.html.

Victor Chernozhukov, Kaspar Wüthrich, and Yinchu Zhu. Distributional conformal prediction. *Proceedings of the National Academy of Sciences*, 118(48):e2107794118, 2021. URL https://doi.org/10.1073/pnas.2107794118.

Yunjie Fan and Matteo Sesia. Interpretable multivariate conformal prediction with fast transductive standardization. *arXiv preprint arXiv:2512.15383*, 2025. URL https://arxiv.org/abs/2512.15383.

Zhenhan Fang, Aixin Tan, and Jian Huang. CONTRA: Conformal prediction region via normalizing flow transformation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=pOO9cqLq7Q.

Isaac Gibbs and Emmanuel J. Candès. Adaptive conformal inference under distribution shift. In *Advances in Neural Information Processing Systems*, volume 34, pp. 1660–1672, 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/0d441de75945e5acbc865406fc9a2559-Paper.pdf.

Tilmann Gneiting and Matthias Katzfuss. Probabilistic forecasting. *Annual Review of Statistics and Its Application*, 1(1):125–151, 2014. URL https://doi.org/10.1146/annurev-statistics-062713-085831.

Jonathan Ho, Ajay N. Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

Tao Hong and Shu Fan. Probabilistic electric load forecasting: A tutorial review. *International Journal of Forecasting*, 32(3):914–938, 2016. URL https://doi.org/10.1016/j.ijforecast.2015.11.011.

Rafael Izbicki, Gilson Shimizu, and Rafael B. Stern. CD-split and HPD-split: Efficient conformal regions in high dimensions. *Journal of Machine Learning Research*, 23(87):1–32, 2022. URL https://www.jmlr.org/papers/v23/20-797.html.

Xiangyu Jiang. Score-based diffusion priors for adaptive conformal inference under distribution shift. In *Proceedings of the 42nd Conference on Uncertainty in Artificial Intelligence*, volume 337 of *Proceedings of Machine Learning Research*, pp. 2517–2537. PMLR, 2026. URL https://proceedings.mlr.press/v337/jiang26a.html.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. URL https://arxiv.org/abs/1412.6980.

Junghwan Lee, Chen Xu, and Yao Xie. Flow-based conformal prediction for multi-dimensional time series. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Uv3efQiPBZ.

Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J. Tibshirani, and Larry Wasserman. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523): 1094–1111, 2018. URL https://doi.org/10.1080/01621459.2017.1307116.

Boris Medina-Salgado, Eddy Sánchez-DelaCruz, Pilar Pozos-Parra, and Javier E Sierra. Urban traffic flow prediction techniques: A review. *Sustainable Computing: Informatics and Systems*, 35: 100739, 2022. URL https://www.sciencedirect.com/science/article/pii/S2210537922000725.

Soundouss Messoudi, Sébastien Destercke, and Sylvain Rousseau. Ellipsoidal conformal inference for multi-target regression. In *Conformal and Probabilistic Prediction with Applications*, pp. 294–306. PMLR, 2022. URL https://proceedings.mlr.press/v179/messoudi22a.html.

Roberto I Oliveira, Paulo Orenstein, Thiago Ramos, and Joao Vitor Romano. Split conformal prediction and non-exchangeable data. *Journal of Machine Learning Research*, 25(225):1–38, 2024. URL http://jmlr.org/papers/v25/23-1553.html.

Pierre Pinson. Very-short-term probabilistic forecasting of wind power with generalized logit–normal distributions. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 61(4): 555–576, 2012. URL https://doi.org/10.1111/j.1467-9876.2011.01026.x.

Javier E Santos and Yen Ting Lin. Using ornstein-uhlenbeck process to understand denoising diffusion probabilistic model and its noise schedules. *arXiv preprint arXiv:2311.17673*, 2023. URL https://arxiv.org/abs/2311.17673.

Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(12):371–421, 2008. URL http://jmlr.org/papers/v9/shafer08a.html.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/3001ef257407d5a371a96dcd947c7d93-Paper.pdf.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. URL https://arxiv.org/abs/2011.13456.

Renukanandan Tumu, Matthew Cleaveland, Rahul Mangharam, George Pappas, and Lars Lindemann. Multi-modal conformal prediction regions by optimizing convex shape templates. In *Proceedings of the 6th Annual Learning for Dynamics and Control Conference*, volume 242 of *Proceedings of Machine Learning Research*, pp. 1343–1356. PMLR, 2024. URL https://proceedings.mlr.press/v242/tumu24a.html.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*. Springer, 2005. URL https://doi.org/10.1007/b106715.

Weijiang Xiong, Robert Fonod, and Nikolas Geroliminis. Unveiling stochasticity: Universal multi-modal probabilistic modeling for traffic forecasting. *Transportation Research Part C: Emerging Technologies*, 191:105854, 2026. URL https://www.sciencedirect.com/science/article/pii/S0968090X26003426.

Chen Xu and Yao Xie. Conformal prediction interval for dynamic time-series. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 11559–11569. PMLR, 2021a. URL https://proceedings.mlr.press/v139/xu21h.html.

Chen Xu and Yao Xie. Conformal anomaly detection on spatio-temporal observations with missing data. *arXiv preprint arXiv:2105.11886*, 2021b. URL https://arxiv.org/abs/2105.11886.

Chen Xu and Yao Xie. Sequential predictive conformal inference for time series. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 38707–38727. PMLR, 2023. URL https://proceedings.mlr.press/v202/xu23r.html.

Chen Xu, Hanyang Jiang, and Yao Xie. Conformal prediction for multi-dimensional time series by ellipsoidal sets. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 55076–55099. PMLR, 2024. URL https://proceedings.mlr.press/v235/xu24m.html.

Minghe Zhang, Chen Xu, Andy Sun, Feng Qiu, and Yao Xie. Solar radiation ramping events modeling using spatio-temporal point processes. *INFORMS Journal on Data Science*, 4(2):173–196, 2025. URL https://doi.org/10.1287/ijds.2023.0006.

APPENDIX A PROOFS IN SECTION 4

APPENDIX A.1 PROOF OF THEOREM 4.1

Proof. Conditional on $v_1, \ldots, v_{|\mathcal{T}|}$, the $|\mathcal{T}|R/2$ base Gaussian perturbations are independent. Assumption 1 therefore implies

$$\mathrm{Var}_\varepsilon \left\{ s_w(x,y) \mid v_1, \ldots, v_{|\mathcal{T}|} \right\} \leq \frac{C(x,y)}{2|\mathcal{T}|R}.$$

The conditional expectation of the empirical score is

$$\mathbb{E}\left[s_w(x,y) \mid v_1, \ldots, v_{|\mathcal{T}|} \right] = \frac{1}{|\mathcal{T}|} \sum_{k=1}^{|\mathcal{T}|} h_{\widehat{\theta},x,y}(v_k).$$

Since v_k is uniformly distributed within an interval of width $1/|\mathcal{T}|$ and $h_{\widehat{\theta},x,y}$ is $L(x,y)$-Lipschitz, the variance can be uniformly bounded by

$$\mathrm{Var}\left\{ h_{\widehat{\theta},x,y}(v_k) \right\} \leq \frac{L^2(x,y)}{2} \mathbb{E}[(v_k - v_k')^2] = L^2(x,y)\mathrm{Var}(v_k) = \frac{L^2(x,y)}{12|\mathcal{T}|^2},$$

where v_k' is i.i.d copy of v_k. Therefore, the independent stratified draws can give

$$\mathrm{Var}_v\left\{\frac{1}{|\mathcal{T}|}\sum_{k=1}^{|\mathcal{T}|} h_{\widehat{\theta},x,y}(v_k)\right\} \le \frac{L^2(x,y)}{12|\mathcal{T}|^3}.$$

Since randomized stratification is unbiased, we have

$$\frac{1}{|\mathcal{T}|}\sum_{k=1}^{|\mathcal{T}|}\mathbb{E}_v\left[h_{\widehat{\theta},x,y}(v_k)\right] = \int_0^1 h_{\widehat{\theta},x,y}(v)\,dv = s_{\widehat{\theta}}(x,y)$$

and by the law of total variance, we have

$$\mathbb{E}_w\left[\left\{s_w(x,y) - s_{\widehat{\theta}}(x,y)\right\}^2\right] \le \frac{C(x,y)}{2|\mathcal{T}|R} + \frac{L^2(x,y)}{12|\mathcal{T}|^3}.$$

The expectation bound follows from Jensen's inequality and the high-probability bound follows from Markov's inequality applied to the squared error. □

APPENDIX A.2 PROOF OF COROLLARY 4.2

Proof of Lemma 4.1. Fix q and let $\mathbb{P}_j$ be the joint law of the first j blocks of that parity, and μ_j the marginal law of its j-th block. By the monotonicity of β,

$$\|\mathbb{P}_j - \mathbb{P}_{j-1}\otimes\mu_j\|_{TV} \le \beta(a+1) \le \beta(a), \qquad 2 \le j \le m. \tag{21}$$

The triangle inequality and invariance of TV imply, by induction,

$$\left\|\mathbb{P}_m - \bigotimes_{j=1}^{m}\mu_j\right\|_{TV} \le (m-1)\beta(a). \tag{22}$$

Therefore, each measurable event $\mathcal{E}_q$ has probability at most its independent-block probability plus $(m-1)\beta(a)$. Use equation 12 in the Assumption 2 and the union bound to obtain

$$\mathbb{P}\{d(x,y,\widehat{\theta}) > r\} \le \sum_{q\in Q}\mathbb{P}(\mathbf{B}^{(q,*)} \in \mathcal{E}_q) + |Q|(m-1)\beta(a) \le \delta_0 + |Q|(m-1)\beta(a). \tag{23}$$

□

Proof of Corollary 4.2. Denote

$$\Delta(v,\varepsilon) = \varepsilon_{\widehat{\theta}}(m_{t(v)}y + \sigma_{t(v)}\varepsilon, t(v), x) - \varepsilon_0(m_{t(v)}y + \sigma_{t(v)}\varepsilon, t(v), x)$$

and

$$\Gamma(v,\varepsilon) = \varepsilon - \varepsilon_0(m_{t(v)}y + \sigma_{t(v)}\varepsilon, t(v), x)$$

Expanding the squared losses and applying the triangle and Cauchy–Schwarz inequalities gives

$$\begin{aligned}|s_{\widehat{\theta}}(x,y) - s_0(x,y)| &\le 2\mathbb{E}_{v,\varepsilon}|\langle\Gamma,\Delta\rangle| + \mathbb{E}_{v,\varepsilon}\|\Delta\|_2^2 \\ &\le 2(\mathbb{E}\|\Gamma\|_2^2)^{1/2}(\mathbb{E}\|\Delta\|_2^2)^{1/2} + \mathbb{E}\|\Delta\|_2^2 \le 2B_0(x,y)r + r^2.\end{aligned}$$

By the triangle inequality,

$$|s_w(x,y) - s_0(x,y)| \le |s_w(x,y) - s_{\widehat{\theta}}(x,y)| + |s_{\widehat{\theta}}(x,y) - s_0(x,y)|$$

and it follows from Lemma 4.1. □

APPENDIX A.3 PROOF OF THEOREM 4.3

Proof. Fix such a $w = \{\varepsilon_{t,r}\}_{r=1}^{R/2}$ from Algorithm 1. Let $Z^* = (X^*, Y^*)$ have the stationary marginal law of Z_1 and be independent of $\mathcal{D}_{\text{train}}$. For any measurable function $q : (\mathcal{X} \times \mathcal{Y})^{n_0} \to \mathbb{R}$, define a threshold on training set $q_{\text{train}} = q(Z_i, i \in I_{\text{train}})$ and the probability under the law $\mathbb{P}_w$ on w

$$P^w_{q,\text{train}} = \mathbb{P}_w(s_w(Z^*) \le q_{\text{train}} \mid \mathcal{D}_{\text{train}}) \tag{24}$$

where the empirical score $s_w(X, Y)$ is constructed from the CRN bank w by

$$s_w(X, Y) = \frac{1}{|\mathcal{T}|R} \sum_{t\in\mathcal{T}} \sum_{r\le R} \|\varepsilon_{t,r} - \hat{\varepsilon}_\theta(m_t Y + \sigma_t \varepsilon_{t,r}), t, X)\|^2.$$

Proposition 13 of Oliveira et al. (2024), applied under $\mathbb{P}_w$ to the variable $1\{s_w(Z_i) \le q_{\text{train}}\}$ for $i \in I_{\text{cal}}$ gives

$$\mathbb{P}_w \left[\left| \frac{1}{n_{\text{cal}}} \sum_{i\in I_{\text{cal}}} 1\{s_w(Z_i) \le q_{\text{train}}\} - P^w_{q,\text{train}} \right| \le \varepsilon_{\text{cal}} \right] \ge 1 - \delta_{\text{cal}} \tag{25}$$

Discard the first $r - 1$ calibration observations to create a gap of length r from the training sample. The total-variation blocking technique changes the probability by at most $\beta(r)$. Split the remaining $n_{\text{cal}} - r + 1 = 2ma$ observations into $2m$ blocks of length a; replacing the odd, and then the even, blocks by independent copies costs at most $4(m-1)\beta(a)$. Bernstein's inequality on the independent blocks uses $\text{Var}(1\{s_w(Z_i) \le q_{\text{train}}\}) \le 1/4$. Stationarity and the covariance bound for bounded β-mixing variables yield the block variance

$$\widetilde{\sigma}^2(a) = \frac{1}{4} + \frac{2}{a} \sum_{j=1}^{a-1} (a - j)\beta(j). \tag{26}$$

For every training-dependent threshold q_{train}, Proposition 14 of Oliveira et al. (2024) gives

$$\left| \mathbb{P}_w \left(s_w(Z_{n+1}) \le q_{\text{train}} \right) - \mathbb{P}_w \left(s_w(Z^\star) \le q_{\text{train}} \right) \right| \le \beta(n_{\text{cal}} + 1). \tag{27}$$

Indeed, replacing the joint law of $\mathcal{D}_{\text{train}}$ and Z_{n+1} by the product of their marginals changes the probability of any measurable event by at most the displayed total-variation distance.

Define the ordinary empirical $(1 - \alpha)$-quantile

$$\widehat{q}^w_{1-\alpha} = \inf \left\{ t \in \mathbb{R} : \frac{1}{n_{\text{cal}}} \sum_{i\in I_{\text{cal}}} 1\{s_w(Z_i) \le t\} \ge 1 - \alpha \right\}. \tag{28}$$

Condition equation 25 and equation 27 are satisfied in Theorem 1 of Oliveira et al. (2024). Therefore, it implies

$$\mathbb{P}_w \left(s_w(Z_{n+1}) \le \widehat{q}^w_{1-\alpha} \right) \ge 1 - \alpha - \varepsilon_{\text{cal}} - \delta_{\text{cal}} - \beta(n_{\text{cal}} + 1). \tag{29}$$

$\widehat{q}^w_{1-\alpha}$ is the $\lceil n_{\text{cal}}(1 - \alpha)\rceil$-th order statistic of the calibration scores, whereas $q^w_{1-\alpha}$ is the $\lceil (n_{\text{cal}} + 1)(1 - \alpha)\rceil$-th order statistic. It is obvious that $q^w_{1-\alpha} \ge \widehat{q}^w_{1-\alpha}$, which implies

$$\left\{ s_w(Z_{n+1}) \le \widehat{q}^w_{1-\alpha} \right\} \subseteq \left\{ s_w(Z_{n+1}) \le q^w_{1-\alpha} \right\}.$$

It follows from equation 29 that

$$\mathbb{P}_w \left(Y_{n+1} \in \widehat{C}^w_{1-\alpha}(X_{n+1}) \right) = \mathbb{P}_w \left(s_w(Z_{n+1}) \le q^w_{1-\alpha} \right) \ge 1 - \alpha - \varepsilon_{\text{cal}} - \delta_{\text{cal}} - \beta(n_{\text{cal}} + 1). \tag{30}$$

□

APPENDIX A.4 PROOF OF THEOREM 4.4

Proof. For any threshold t and fixed bank w, given the training sample, for $A \in \mathcal{A}_\gamma$, define

$$P^w_{\text{train}}(t; A) := \mathbb{P}_w \left(s_w(Z^*) \le t \mid \mathcal{D}_{\text{train}}, X^* \in A \right). \tag{31}$$

where $Z^* = (X^*, Y^*)$ has the stationary marginal distribution of Z_i and is independent of the observed data, the CRN banks, and any training randomness. Conditioning on $\mathcal{D}_{\text{train}}$ also includes any independent randomness used to fit the score.

By Lemma 22 and the blocking argument in Proposition 23 of Oliveira et al. (2024), for any $(a, m, r) \in \mathcal{G}_{\text{cal}}$, the first $r-1$ calibration observations are discarded to create a separation between the training and calibration samples. The remaining $n_{\text{cal}} - r + 1 = 2ma$ observations are divided into $2m$ blocks of length a. The blocking coupling contributes the mixing terms $16(m-1)\beta(a) + \beta(r)$, while the VC bound for $\mathcal{A}$, together with the lower probability bound γ gives ε_{cal} in 18. Under the decoupled product law, the retained calibration block is independent of the training sample. Conditional on the training sample and its training randomness, the fixed bank w makes s_w deterministic, so Lemma 22 applies uniformly over A and t. Therefore,

$$\mathbb{P}_w \left[\sup_{A \in \mathcal{A}_\gamma, t \in \mathbb{R}} \left| \frac{1}{\max\{n_{\text{cal}}(A), 1\}} \sum_{i \in I_{\text{cal}}(A)} 1\{s_w(Z_i) \le t\} - P^w_{\text{train}}(t; A) \right| \le \varepsilon_{\text{cal}} \right] \ge 1 - \delta_{\text{cal}}, \tag{32}$$

where $n_{\text{cal}}(A) = |I_{\text{cal}}(A)|$ with $I_{\text{cal}}(A) := \{i \in I_{\text{cal}} : X_i \in A\}$.

Fix w and given the training sample $\mathcal{D}_{\text{train}}$, for $\phi \in (0, 1)$, define the conditional population quantile

$$q^w_{\phi,\text{train}}(A) = \inf \left\{ t \in \mathbb{R} : \mathbb{P}_w \left(s_w(Z^*) \le t \mid \mathcal{D}_{\text{train}}, X^* \in A \right) \ge \phi \right\}, \tag{33}$$

and define the ordinary empirical conditional quantile

$$\tilde{q}^w_{1-\alpha}(A) = \inf \left\{ t \in \mathbb{R} : \frac{1}{\max\{n_{\text{cal}}(A), 1\}} \sum_{i \in I_{\text{cal}}(A)} 1\{s_w(Z_i) \le t\} \ge 1 - \alpha \right\}, \tag{34}$$

and $\tilde{q}^w_{1-\alpha}(A) = +\infty$ when the number does not exist.

If $1 - \alpha - \varepsilon_{\text{cal}} \le 0$, the claimed lower bound is nonpositive and the result is trivial. Otherwise, let $\phi = 1 - \alpha - \varepsilon_{\text{cal}} \in (0, 1)$.

By the definition of equation 33, for each $l > 0$

$$\mathbb{P}_w(s_w(Z^*) \le q^w_{1-\alpha-\varepsilon_{\text{cal}},\text{train}}(A) - 1/l \mid \mathcal{D}_{\text{train}}, X^* \in A) < 1 - \alpha - \varepsilon_{\text{cal}}. \tag{35}$$

Therefore, given the event in equation 32,

$$\frac{1}{\max\{n_{\text{cal}}(A), 1\}} \sum_{i \in I_{\text{cal}}(A)} 1\{s_w(Z_i) \le q^w_{1-\alpha-\varepsilon_{\text{cal}},\text{train}}(A) - 1/l\} < 1 - \alpha.$$

By the definition of $\tilde{q}^w_{1-\alpha}(A)$, it implies $q^w_{1-\alpha-\varepsilon_{\text{cal}},\text{train}}(A) - 1/l \le \tilde{q}^w_{1-\alpha}(A)$. Letting $l \to \infty$ gives $q^w_{1-\alpha-\varepsilon_{\text{cal}},\text{train}}(A) \le \tilde{q}^w_{1-\alpha}(A)$. Define the calibration event

$$\mathcal{E}_{\text{cal}} := \{q^w_{1-\alpha-\varepsilon_{\text{cal}},\text{train}}(A) \le \tilde{q}^w_{1-\alpha}(A) \text{ for all } A \in \mathcal{A}_\gamma\}. \tag{36}$$

We have $\mathbb{P}_w(\mathcal{E}^c_{\text{cal}}) \le \delta_{\text{cal}}$.

The conditional decoupling result in Proposition 24 of Oliveira et al. (2024) gives, for each $q_{train}(A)$,

$$\begin{aligned} &\Big| \mathbb{P}_w \left(s_w(Z_{n+1}) \le q_{\text{train}}(A) \mid X_{n+1} \in A \right) \\ &\quad - \mathbb{P}_w \left(s_w(Z^*) \le q_{\text{train}}(A) \mid X^* \in A \right) \Big| \le \frac{\beta(n_{\text{cal}} + 1)}{\gamma}. \end{aligned} \tag{37}$$

Given the event $\mathcal{E}_{\text{cal}}$, monotonicity implies

$$\{s_w(Z_{n+1}) \le q^w_{1-\alpha-\varepsilon_{\text{cal}},\text{train}}(A)\} \cap \mathcal{E}_{\text{cal}} \subseteq \{s_w(Z_{n+1}) \le \tilde{q}^w_{1-\alpha}(A)\}. \tag{38}$$

Therefore, combining equation 32, equation 37, we have

$$\begin{aligned}&\mathbb{P}_w(s_w(Z_{n+1}) \le \tilde{q}^w_{1-\alpha}(A) \mid X_{n+1} \in A)\\ &\ge \mathbb{P}_w(s_w(Z_{n+1}) \le q^w_{1-\alpha-\varepsilon_{\text{cal}},\text{train}}(A) \mid X_{n+1} \in A) - \mathbb{P}_w(\mathcal{E}^c_{\text{cal}} \mid X_{n+1} \in A)\\ &\ge 1-\alpha-\varepsilon_{\text{cal}} - \beta(n_{\text{cal}}+1)/\gamma - \delta_{\text{cal}}/\gamma.\end{aligned} \tag{39}$$

Finally, let $q^w_{1-\alpha}(A)$ be the $\lceil (n_{\text{cal}}(A)+1)(1-\alpha)\rceil$-th order statistic of the scores $\{s_w(Z_i) : i \in I_{\text{cal}}(A)\}$ and $q^w_{1-\alpha}(A) = +\infty$ when $\lceil (n_{\text{cal}}(A)+1)(1-\alpha)\rceil > n_{\text{cal}}(A)$. Since $q^w_{1-\alpha}(A) \ge \tilde{q}^w_{1-\alpha}(A)$, we have

$$\{s_w(Z_{n+1}) \le \tilde{q}^w_{1-\alpha}(A)\} \subseteq \{s_w(Z_{n+1}) \le q^w_{1-\alpha}(A)\}.$$

Therefore,

$$\begin{aligned}\mathbb{P}_w(Y_{n+1} \in \widehat{C}^w_{1-\alpha}(X_{n+1}; A) \mid X_{n+1} \in A) &= \mathbb{P}_w\left(s_w(Z_{n+1}) \le q^w_{1-\alpha}(A) \mid X_{n+1} \in A\right)\\ &\ge 1-\alpha-\varepsilon_{\text{cal}} - \delta_{\text{cal}}/\gamma - \beta(n_{\text{cal}}+1)/\gamma,\end{aligned}$$

which proves the result.

□

APPENDIX B EXPERIMENT DETAILS

APPENDIX B.1 EXPERIMENT SETUP IN BOTH NUMERICAL AND REAL-WORLD DATASETS

Residual variant. We additionally consider a residual-based variant that separates conditional-mean prediction from uncertainty modeling. Specifically, we estimate a deterministic point predictor $\widehat{f}$ using only $\mathcal{D}_{\text{train}}$ and train the score model on the residuals $e_i = Y_i - \widehat{f}(X_i)$. For a fixed randomized bank w, the residual nonconformity score is

$$s_w(x,y) = \frac{1}{|\mathcal{T}|R}\sum_{k=1}^{|\mathcal{T}|}\sum_{r=1}^{R}\left\|\varepsilon_{k,r} - \widehat{\varepsilon}_\theta\left(m_{t_k}\big(y-\widehat{f}(x)\big) + \sigma_{t_k}\varepsilon_{k,r}, t_k, x\right)\right\|_2^2. \tag{40}$$

The same fixed evaluation bank and split-conformal calibration procedure as in Algorithm 1 are used. The Algorithm 2 follows below.

FlowCP We implement the method using the source code released by the authors (Lee et al., 2026). Table 2 shows the hyperparameter search space and Table 3 shows the optimized hyperparameter configuration. For the numerical simulation, the context window sizes for the encoder and LSTM/LOO bootstrap are set to 1 since the dimension of oracle covariates is 1. For the real data study, the context window sizes for the encoder and LSTM/LOO bootstrap are set to 50. We train the model using Adam (Kingma & Ba, 2014) for 50 epochs with dropout for all experiments and use the validation set to select the best model.

Table 2: The hyperparameter search space for FlowCP

	Hyperparameter	Search space
Vector field	the number of layers	$\{2, 4, 6\}$
	hidden dimension	$\{16, 32, 64\}$
Encoder	the number of layers	$\{2, 4, 6\}$
	the number of heads	$\{2, 4, 8\}$
	model dimension	$\{16, 32, 64\}$
	dropout	$\{0, 0.1\}$
General	covariance scale γ	$\{1, 2, 4, 8\}$
	learning rate	$\{0.0005, 0.0001\}$
	batch size	$\{8, 16\}$

MultiDimSPCI We implement MultiDimSPCI (Xu et al., 2024) using the source code released by the authors. The context window size is set to 50 for all real-world datasets, consistent with the setup used for FlowCP. Following the original publication, the number of trees for the quantile random forest is set to 15. The training and validation sets are combined into a single training set.

Algorithm 2 Residual conformal prediction via conditional score-based SDE denoising

Require: Time series data $\mathcal{D} = \{(X_i, Y_i)\}_{i=1}^{n}$; split point n_0; test context X_{n+1}; $\alpha \in (0,1)$; kernel (m_t, σ_t); $0 < \delta < T$; invertible decreasing log-SNR λ; time-grid size $|\mathcal{T}|$; even number $R \geq 2$.

1: Set $\mathcal{D}_{\text{train}} = \{(X_i, Y_i)\}_{i=1}^{n_0}$ and $n_{\text{cal}} \leftarrow n - n_0$.

2: Train the deterministic predictor $\widehat{f}$ using only $\mathcal{D}_{\text{train}}$.

3: Compute the training residuals $e_i \leftarrow Y_i - \widehat{f}(X_i)$, $i = 1, \ldots, n_0$.

4: Train $\widehat{\varepsilon}_\theta$ on $\{(X_i, e_i)\}_{i=1}^{n_0}$ by minimizing

$$\widehat{\theta} \in \arg\min_{\theta} \frac{1}{n_0} \sum_{i=1}^{n_0} \mathbb{E}_{v,\varepsilon} \left[\left\| \varepsilon - \varepsilon_\theta \left(m_{t(v)} e_i + \sigma_{t(v)} \varepsilon, t(v), X_i \right) \right\|_2^2 \right],$$

where $v \sim \mathcal{U}(0,1)$, $\varepsilon \sim \mathcal{N}(0, I_{d_y})$, and $t(v) = \lambda^{-1}\left((1-v)\lambda(\delta) + v\lambda(T)\right)$.

5: **for** $k = 1, \ldots, |\mathcal{T}|$ **do**

6: Draw $U_k \sim \mathcal{U}(0,1)$ independently and set $v_k \leftarrow \frac{k-1+U_k}{|\mathcal{T}|}$.

7: Set $\lambda_k \leftarrow (1-v_k)\lambda(\delta) + v_k\lambda(T)$ and $t_k \leftarrow \lambda^{-1}(\lambda_k)$.

8: Draw independent $\varepsilon_{k,r} \sim \mathcal{N}(0, I_{d_y})$, $r = 1, \ldots, R/2$, and set $\varepsilon_{k,r+R/2} \leftarrow -\varepsilon_{k,r}$.

9: **end for**

10: Freeze $w = \{t_k, \varepsilon_{k,r} : k \leq |\mathcal{T}|,\ r \leq R\}$, generated independently of the data and the training procedure.

11: Compute $s_i = s_w(X_i, Y_i)$ for $i = n_0 + 1, \ldots, n$ using equation 40.

12: Set $\widehat{q}_{1-\alpha}$ by equation 8, including the $+\infty$ convention.

13: **return** $\widehat{\mathcal{C}}_{1-\alpha}(X_{n+1})$ in equation 9, using the same w for all candidates and score in equation 40.

Table 3: The optimized hyperparameter configuration for FlowCP based on the hyperparameter grid search.

Dataset	Hyperparameter	$d_y = 2$	$d_y = 4$
Traffic	the number of layers of the vector field	4	4
	the number of heads of the encoder	2	2
	the number of layers of the encoder	4	4
	the hidden dimension of the vector field and encoder	32	32
	covariance scale γ	1	1
	encoder dropout	0.1	0.1
	batch size	8	8
	learning rate	0.0001	0.0001
	null condition probability	0.05	0.05
	guidance scale w (LOO/LSTM base predictor)	1.1/1.2	1.1/1.2
Solar	the number of layers of the vector field	4	4
	the number of heads of the encoder	2	2
	the number of layers of the encoder	4	4
	the hidden dimension of the vector field and encoder	32	32
	covariance scale γ	1	1
	encoder dropout	0.1	0.1
	batch size	8	8
	learning rate	0.0005	0.0005
	null condition probability	0.05	0.05
	guidance scale w (LOO/LSTM base predictor)	1.5/1.2	1.2/1.1

CONTRA We implement CONTRA (Fang et al., 2025) following the methodology and details provided in the original paper, which does not release any source code. We use six coupling layers with a hidden dimension of 128 and train the model for 100 epochs with the same batch size as FCP and a learning rate of 0.001. The training and validation sets are combined into a calibration set, of which 50% is used to train the model and the remaining 50% is used to calibrate the prediction sets.

Mahalanobis ellipsoid We implement a global Mahalanobis ellipsoid as a shape-restricted conformal baseline. A multi-output linear predictor is estimated on the proper training set, and a single covariance matrix is estimated from its training residuals. We then use the squared Mahalanobis distance induced by this covariance matrix as the nonconformity score and calibrate the ellipsoid radius on the held-out calibration set.

APPENDIX B.2 EXPERIMENT DESIGN FOR NUMERICAL STUDIES

Objective. We evaluate whether a score-based conformal predictor can construct efficient prediction regions under both unimodal and multimodal innovation distributions. The details of other methods are provided in Appendix B.1. All methods are calibrated to the same marginal coverage level, so the primary comparison concerns region volume.

Data-generating process. We generate a stable bivariate VAR(1) process according to $X_t = AX_{t-1} + \varepsilon_t$, where the transition matrix is fixed as $A = \begin{pmatrix} 0.5 & -0.3 \\ 0.2 & 0.4 \end{pmatrix}$. The spectral radius satisfies $\rho(A) = \sqrt{0.26} \approx 0.510 < 1$, which ensures that the data-generating process is stable. The same transition matrix is used for every innovation distribution and random seed so that differences between experimental settings arise only from the innovation law. The first 500 observations are discarded as burn-in.

For the Gaussian baseline, the innovations follow $\varepsilon_t \sim \mathcal{N}\left(0, \begin{pmatrix} 1 & 0.3 \\ 0.3 & 1 \end{pmatrix}\right)$. For the multimodal experiments, the innovations follow an equally weighted six-component Gaussian mixture, $p_\varepsilon(e) = 1/6 \sum_{j=1}^{6} \varphi_2(e; \mu_j, \sigma^2 I_2)$, where $\varphi_2(\cdot; \mu, \Sigma)$ denotes the bivariate Gaussian density and the component centers are equally spaced on a ring, $\mu_j = r\left(\cos\left(2\pi(j-1)/6\right), \sin\left(2\pi(j-1)/6\right)\right)^\top$. For this six-component arrangement, the distance between adjacent centers equals the ring radius, $\delta = \|\mu_{j+1} - \mu_j\|_2 = r$. We consider two standardized separation levels, $\frac{\delta}{\sigma} \in \{5.0, 8.3\}$. The 5.0σ setting evaluates moderately separated modes whose high-coverage level sets may merge, whereas the 8.3σ setting evaluates whether a method can recover six clearly separated components.

Chronological data splitting. Each realization is divided into three contiguous chronological blocks, $I_{\text{train}} = \{1, \ldots, n_0\}$, $I_{\text{cal}} = \{n_0 + 1, \ldots, n\}$, $I_{\text{test}} = \{n + 1, \ldots, n + n_{test}\}$. No observations are shuffled or transferred between blocks. Model estimation, hyperparameter selection, and density estimation use only the training block. The calibration block is used only to choose conformal thresholds, and the final coverage and efficiency metrics are computed only on the test block.

Coverage evaluation. Out-of-sample empirical coverage given $X_i, i \in I_{\text{test}}$ is defined as

$$CVR_{1-\alpha} = \frac{1}{n_{\text{test}}} \sum_{i \in I_{\text{test}}} 1\left\{Y_i \in \widehat{C}_{1-\alpha}(X_i)\right\}.$$

We evaluate $\alpha \in \{0.05, 0.10\}$, corresponding to nominal coverage levels of 0.95 and 0.90. Most methods achieve empirical coverage close to the nominal levels. We interpret prediction-region volumes alongside empirical coverage to account for differences in achieved coverage.

Randomized quasi-Monte Carlo volume estimation. A bounding box $\mathcal{B}_{1-\alpha}$ is expanded until the estimated region is bounded and does not intersect the boundary of the box. For randomized quasi-Monte Carlo points $U_1, \ldots, U_M$ uniformly distributed over this box, volume is estimated by

$$\widehat{V}_{1-\alpha} = \frac{\text{Vol}(\mathcal{B}_{1-\alpha})}{M} \sum_{\ell=1}^{M} \mathbf{1}\left\{U_\ell \in \widehat{C}_{1-\alpha}(X_i)\right\}.$$

Independent random scramblings are used to estimate the numerical integration uncertainty. If $\widehat{V}_{1-\alpha,b}$ denotes the estimate from scramble b, the quadrature standard error is

$$\text{SE}_{\text{QMC}} = \sqrt{\frac{1}{B(B-1)} \sum_{b=1}^{B} \left(\widehat{V}_{1-\alpha,b} - \overline{V}_{1-\alpha}\right)^2},$$

where $\overline{V}_{1-\alpha} = \frac{1}{B}\sum_{b=1}^{B} \widehat{V}_{1-\alpha,b}$ denotes the mean of the estimates.

Result. Across all three innovation distributions, DISCO closely tracks the nominal coverage levels: the empirical coverage of its variants remains approximately 95% and 90%. This shows that the differences in volume primarily reflect prediction-region efficiency rather than a coverage/volume trade-off.

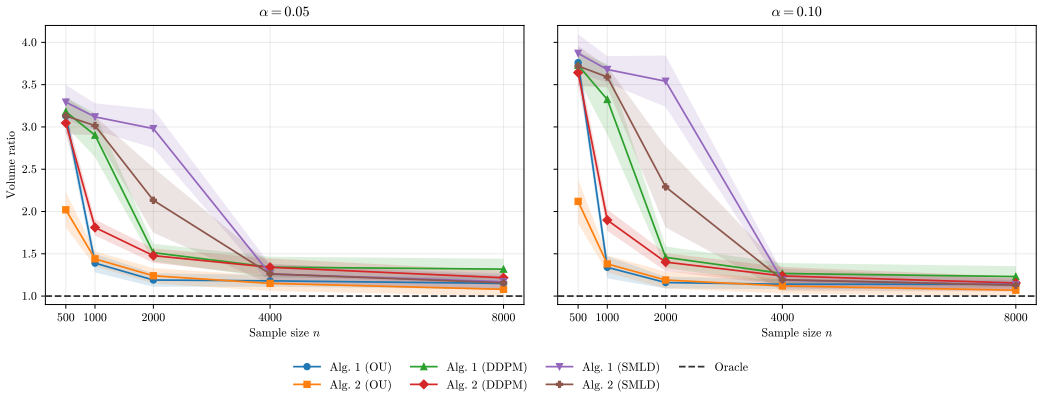


Figure 2: Normalized prediction-region volume on the VAR(1) process with spherical Gaussian-mixture innovations separated by 8.3σ for different sample size n. The left and right panels correspond to $\alpha = 0.05$ and $\alpha = 0.10$, respectively. Lines mean the average volume ratio over 30 seeds, and shaded regions indicate $\pm$ one standard deviation.

Table 6 and Figure 2 show the decrease in volume ratio to the oracle prediction region area when the sample size n increases. DISCO reduces the area by 39.7% and 45.6%, respectively, relative to the smallest baseline area. In contrast, even though CONTRA and MultiDimSPCI can extract basic shapes, they provide larger volumes. The performance of FlowCP usually depends on the length of the context window. That's why it can obtain poor performance in the numerical studies under a context window length of 1 in two different predictors and encoders, even though it can achieve a higher coverage level and relatively efficient volume in the real-world datasets.

Table 5 shows that prediction-region volume ratios generally decrease with sample size in the 5.0σ mixture setting, and 8.3σ setting shows similar trend. 4 shows that DISCO produces volumes close to the oracle volume under Gaussian innovations, particularly for the residual variant. The Mahalanobis baseline is well specified in this setting and generally achieves volumes close to the oracle.

Appendix B.3 Dataset details

Traffic dataset The traffic dataset includes traffic flow collected at 15 different locations (Xu & Xie, 2021b). Each location provides 8878 observations with 5 features. We randomly choose $d_y \in \{2, 4\}$ locations to establish 5 sequences of d_y-dimensional time series. We did not construct sequences with $d_y = 8$ due to the limited number of unique locations, which could lead to overlapping sequences across different trials of experiments.

Solar dataset The solar dataset records solar radiation in Diffused Horizontal Irradiance (DHI) units at 9 different solar sensor locations (Zhang et al., 2025). Each location provides 8755 records with 5 features. We randomly choose $d_y \in \{2, 4\}$ locations to establish 5 sequences of d_y-dimensional time series. We did not construct sequences with $d_y = 8$ due to the limited number of unique locations, which could lead to overlapping sequences across different trials of experiments.

Appendix B.4 Volume estimation in the real data study

Because the prediction regions may be non-convex or disconnected, we estimate their volume using importance sampling rather than integration over a finite bounding box. For a full-support proposal

Table 4: VAR(1) prediction regions under Gaussian innovation. CVR denotes empirical coverage and volume is normalized by the oracle highest-density region. Results are mean $\pm$ one standard deviation over 30 seeds. **Bold** and underlining denote the smallest and second-smallest mean volumes, respectively, for each n and α.

		$\alpha = 0.05$		$\alpha = 0.10$	
n	Method	CVR (%)	Volume Ratio	CVR (%)	Volume Ratio
500	Ours (Algorithm 1, OU)	$94.9_{\pm 0.5}$	$1.037_{\pm 0.034}$	$89.9_{\pm 0.8}$	$1.038_{\pm 0.035}$
	Ours (Algorithm 1, DDPM)	$94.8_{\pm 0.6}$	$1.033_{\pm 0.038}$	$89.9_{\pm 0.7}$	$1.039_{\pm 0.033}$
	Ours (Algorithm 1, SMLD)	$95.0_{\pm 0.5}$	$1.033_{\pm 0.033}$	$90.0_{\pm 0.7}$	$1.039_{\pm 0.032}$
	Ours (Algorithm 2, OU)	$95.0_{\pm 0.6}$	$1.016_{\pm 0.035}$	$89.9_{\pm 0.9}$	$\underline{1.013_{\pm 0.030}}$
	Ours (Algorithm 2, DDPM)	$94.9_{\pm 0.6}$	$\underline{1.015_{\pm 0.039}}$	$90.0_{\pm 0.9}$	$1.015_{\pm 0.031}$
	Ours (Algorithm 2, SMLD)	$95.0_{\pm 0.5}$	$1.024_{\pm 0.033}$	$90.0_{\pm 0.8}$	$1.023_{\pm 0.029}$
	FlowCP (LSTM + encoder)	$94.9_{\pm 0.5}$	$1.134_{\pm 0.065}$	$89.9_{\pm 0.7}$	$1.123_{\pm 0.063}$
	FlowCP (LOO bootstrap + encoder)	$94.9_{\pm 0.5}$	$1.083_{\pm 0.029}$	$90.0_{\pm 0.6}$	$1.072_{\pm 0.028}$
	MultiDimSPCI (LSTM)	$93.8_{\pm 0.5}$	$1.122_{\pm 0.083}$	$89.6_{\pm 0.3}$	$1.201_{\pm 0.090}$
	MultiDimSPCI (LOO bootstrap)	$94.1_{\pm 0.4}$	$1.050_{\pm 0.021}$	$89.7_{\pm 0.2}$	$1.112_{\pm 0.018}$
	CONTRA (LSTM)	$94.9_{\pm 0.7}$	$1.169_{\pm 0.224}$	$89.9_{\pm 0.8}$	$1.119_{\pm 0.143}$
	CONTRA (LOO bootstrap)	$94.9_{\pm 0.8}$	$1.021_{\pm 0.047}$	$90.1_{\pm 0.8}$	$1.024_{\pm 0.033}$
	Mahalanobis ellipsoid	$95.0_{\pm 0.6}$	$\mathbf{1.011_{\pm 0.031}}$	$90.0_{\pm 0.8}$	$\mathbf{1.011_{\pm 0.030}}$
1000	Ours (Algorithm 1, OU)	$94.9_{\pm 0.7}$	$1.026_{\pm 0.043}$	$90.1_{\pm 0.8}$	$1.029_{\pm 0.031}$
	Ours (Algorithm 1, DDPM)	$94.9_{\pm 0.6}$	$1.027_{\pm 0.039}$	$90.0_{\pm 0.8}$	$1.023_{\pm 0.030}$
	Ours (Algorithm 1, SMLD)	$95.0_{\pm 0.5}$	$1.028_{\pm 0.036}$	$90.0_{\pm 0.8}$	$1.027_{\pm 0.030}$
	Ours (Algorithm 2, OU)	$95.0_{\pm 0.6}$	$1.009_{\pm 0.034}$	$90.0_{\pm 0.9}$	$1.012_{\pm 0.030}$
	Ours (Algorithm 2, DDPM)	$95.0_{\pm 0.6}$	$\underline{1.008_{\pm 0.036}}$	$90.0_{\pm 0.9}$	$\underline{1.010_{\pm 0.030}}$
	Ours (Algorithm 2, SMLD)	$94.9_{\pm 0.6}$	$1.015_{\pm 0.034}$	$90.0_{\pm 0.8}$	$1.017_{\pm 0.028}$
	FlowCP (LSTM + encoder)	$95.0_{\pm 0.5}$	$1.102_{\pm 0.037}$	$89.9_{\pm 0.7}$	$1.083_{\pm 0.038}$
	FlowCP (LOO bootstrap + encoder)	$95.0_{\pm 0.5}$	$1.065_{\pm 0.038}$	$89.9_{\pm 0.6}$	$1.054_{\pm 0.039}$
	MultiDimSPCI (LSTM)	$94.0_{\pm 0.4}$	$1.063_{\pm 0.032}$	$89.7_{\pm 0.2}$	$1.126_{\pm 0.029}$
	MultiDimSPCI (LOO bootstrap)	$94.1_{\pm 0.4}$	$1.049_{\pm 0.021}$	$89.8_{\pm 0.2}$	$1.112_{\pm 0.016}$
	CONTRA (LSTM)	$94.8_{\pm 0.6}$	$1.026_{\pm 0.059}$	$89.8_{\pm 0.8}$	$1.022_{\pm 0.048}$
	CONTRA (LOO bootstrap)	$94.9_{\pm 0.7}$	$1.016_{\pm 0.044}$	$89.9_{\pm 0.9}$	$1.011_{\pm 0.033}$
	Mahalanobis ellipsoid	$94.9_{\pm 0.6}$	$\mathbf{1.001_{\pm 0.032}}$	$90.1_{\pm 0.8}$	$\mathbf{1.008_{\pm 0.029}}$
2000	Ours (Algorithm 1, OU)	$95.0_{\pm 0.6}$	$1.017_{\pm 0.041}$	$90.0_{\pm 0.8}$	$1.015_{\pm 0.030}$
	Ours (Algorithm 1, DDPM)	$95.0_{\pm 0.6}$	$1.016_{\pm 0.034}$	$90.0_{\pm 0.8}$	$1.011_{\pm 0.027}$
	Ours (Algorithm 1, SMLD)	$95.0_{\pm 0.6}$	$1.015_{\pm 0.033}$	$89.9_{\pm 0.8}$	$1.014_{\pm 0.028}$
	Ours (Algorithm 2, OU)	$95.0_{\pm 0.6}$	$1.007_{\pm 0.034}$	$89.9_{\pm 0.8}$	$1.005_{\pm 0.027}$
	Ours (Algorithm 2, DDPM)	$95.0_{\pm 0.6}$	$\underline{1.005_{\pm 0.032}}$	$89.9_{\pm 0.8}$	$1.004_{\pm 0.029}$
	Ours (Algorithm 2, SMLD)	$95.0_{\pm 0.7}$	$1.009_{\pm 0.036}$	$89.9_{\pm 0.8}$	$1.006_{\pm 0.027}$
	FlowCP (LSTM + encoder)	$94.9_{\pm 0.5}$	$1.076_{\pm 0.027}$	$90.1_{\pm 0.6}$	$1.067_{\pm 0.027}$
	FlowCP (LOO bootstrap + encoder)	$95.0_{\pm 0.6}$	$1.053_{\pm 0.056}$	$90.0_{\pm 0.7}$	$1.042_{\pm 0.041}$
	MultiDimSPCI (LSTM)	$94.0_{\pm 0.5}$	$1.056_{\pm 0.019}$	$89.8_{\pm 0.2}$	$1.118_{\pm 0.016}$
	MultiDimSPCI (LOO bootstrap)	$93.8_{\pm 0.5}$	$1.047_{\pm 0.021}$	$89.8_{\pm 0.2}$	$1.113_{\pm 0.016}$
	CONTRA (LSTM)	$94.9_{\pm 0.6}$	$1.012_{\pm 0.041}$	$89.8_{\pm 0.7}$	$1.009_{\pm 0.023}$
	CONTRA (LOO bootstrap)	$94.9_{\pm 0.6}$	$1.008_{\pm 0.038}$	$89.9_{\pm 0.8}$	$\underline{1.004_{\pm 0.034}}$
	Mahalanobis ellipsoid	$95.0_{\pm 0.6}$	$\mathbf{1.000_{\pm 0.033}}$	$90.0_{\pm 0.8}$	$\mathbf{1.001_{\pm 0.029}}$
4000	Ours (Algorithm 1, OU)	$95.0_{\pm 0.6}$	$1.012_{\pm 0.033}$	$90.0_{\pm 0.7}$	$1.012_{\pm 0.026}$
	Ours (Algorithm 1, DDPM)	$95.0_{\pm 0.6}$	$1.017_{\pm 0.034}$	$90.0_{\pm 0.8}$	$1.016_{\pm 0.031}$
	Ours (Algorithm 1, SMLD)	$95.0_{\pm 0.6}$	$1.009_{\pm 0.034}$	$90.0_{\pm 0.8}$	$1.010_{\pm 0.029}$
	Ours (Algorithm 2, OU)	$95.0_{\pm 0.6}$	$1.005_{\pm 0.030}$	$89.9_{\pm 0.8}$	$1.001_{\pm 0.026}$
	Ours (Algorithm 2, DDPM)	$95.0_{\pm 0.6}$	$1.008_{\pm 0.034}$	$89.9_{\pm 0.8}$	$1.002_{\pm 0.027}$
	Ours (Algorithm 2, SMLD)	$95.0_{\pm 0.6}$	$1.005_{\pm 0.035}$	$90.0_{\pm 0.8}$	$1.004_{\pm 0.029}$
	FlowCP (LSTM + encoder)	$94.9_{\pm 0.5}$	$1.056_{\pm 0.037}$	$90.1_{\pm 0.8}$	$1.049_{\pm 0.039}$
	FlowCP (LOO bootstrap + encoder)	$95.0_{\pm 0.6}$	$1.081_{\pm 0.122}$	$89.9_{\pm 0.7}$	$1.051_{\pm 0.076}$
	MultiDimSPCI (LSTM)	$93.8_{\pm 0.6}$	$1.054_{\pm 0.021}$	$89.8_{\pm 0.2}$	$1.114_{\pm 0.014}$
	MultiDimSPCI (LOO bootstrap)	$93.7_{\pm 0.6}$	$1.053_{\pm 0.023}$	$89.9_{\pm 0.2}$	$1.113_{\pm 0.015}$
	CONTRA (LSTM)	$94.9_{\pm 0.7}$	$1.004_{\pm 0.035}$	$89.9_{\pm 0.8}$	$1.004_{\pm 0.028}$
	CONTRA (LOO bootstrap)	$94.9_{\pm 0.6}$	$\underline{1.001_{\pm 0.028}}$	$89.8_{\pm 0.8}$	$\mathbf{0.998_{\pm 0.024}}$
	Mahalanobis ellipsoid	$95.0_{\pm 0.6}$	$\mathbf{1.000_{\pm 0.035}}$	$89.9_{\pm 0.8}$	$\underline{0.998_{\pm 0.026}}$
8000	Ours (Algorithm 1, OU)	$94.9_{\pm 0.6}$	$1.007_{\pm 0.034}$	$90.0_{\pm 0.8}$	$1.008_{\pm 0.029}$
	Ours (Algorithm 1, DDPM)	$94.9_{\pm 0.6}$	$1.006_{\pm 0.031}$	$90.0_{\pm 0.7}$	$1.009_{\pm 0.026}$
	Ours (Algorithm 1, SMLD)	$95.0_{\pm 0.6}$	$1.010_{\pm 0.032}$	$90.0_{\pm 0.8}$	$1.009_{\pm 0.028}$
	Ours (Algorithm 2, OU)	$95.0_{\pm 0.7}$	$1.005_{\pm 0.036}$	$90.0_{\pm 0.8}$	$1.002_{\pm 0.027}$
	Ours (Algorithm 2, DDPM)	$95.0_{\pm 0.6}$	$1.007_{\pm 0.036}$	$89.9_{\pm 0.8}$	$\underline{1.001_{\pm 0.028}}$
	Ours (Algorithm 2, SMLD)	$95.1_{\pm 0.6}$	$1.008_{\pm 0.034}$	$90.0_{\pm 0.8}$	$1.004_{\pm 0.029}$
	FlowCP (LSTM + encoder)	$95.0_{\pm 0.5}$	$1.094_{\pm 0.191}$	$89.9_{\pm 0.8}$	$1.065_{\pm 0.131}$
	FlowCP (LOO bootstrap + encoder)	$94.9_{\pm 0.6}$	$1.292_{\pm 0.261}$	$89.9_{\pm 0.8}$	$1.173_{\pm 0.158}$
	MultiDimSPCI (LSTM)	$93.6_{\pm 0.6}$	$1.050_{\pm 0.024}$	$89.8_{\pm 0.3}$	$1.110_{\pm 0.011}$
	MultiDimSPCI (LOO bootstrap)	$93.2_{\pm 0.7}$	$1.049_{\pm 0.017}$	$89.9_{\pm 0.4}$	$1.111_{\pm 0.012}$
	CONTRA (LSTM)	$95.0_{\pm 0.5}$	$1.007_{\pm 0.028}$	$90.0_{\pm 0.6}$	$1.006_{\pm 0.020}$
	CONTRA (LOO bootstrap)	$95.0_{\pm 0.5}$	$\underline{1.005_{\pm 0.027}}$	$90.0_{\pm 0.6}$	$1.004_{\pm 0.021}$
	Mahalanobis ellipsoid	$95.0_{\pm 0.6}$	$\mathbf{1.000_{\pm 0.033}}$	$90.0_{\pm 0.8}$	$\mathbf{0.998_{\pm 0.027}}$

Table 5: VAR(1) prediction regions under spherical Gaussian-mixture innovations with 5.0σ separation. CVR denotes empirical coverage and volume is normalized by the oracle highest-density region. Results are mean $\pm$ one standard deviation over 30 seeds. **Bold** and underlining denote the smallest and second-smallest mean volumes, respectively, for each n and α.

		$\alpha = 0.05$		$\alpha = 0.10$	
n	Method	CVR (%)	Volume Ratio	CVR (%)	Volume Ratio
500	Ours (Algorithm 1, OU)	$94.8_{\pm 0.5}$	$1.425_{\pm 0.033}$	$89.9_{\pm 0.7}$	$1.597_{\pm 0.046}$
	Ours (Algorithm 1, DDPM)	$94.8_{\pm 0.5}$	$1.433_{\pm 0.042}$	$89.6_{\pm 0.7}$	$1.585_{\pm 0.043}$
	Ours (Algorithm 1, SMLD)	$94.8_{\pm 0.6}$	$1.466_{\pm 0.066}$	$89.8_{\pm 0.7}$	$1.618_{\pm 0.067}$
	Ours (Algorithm 2, OU)	$94.9_{\pm 0.5}$	$\underline{1.342}_{\pm 0.073}$	$90.0_{\pm 0.8}$	$\underline{1.359}_{\pm 0.089}$
	Ours (Algorithm 2, DDPM)	$94.9_{\pm 0.6}$	$1.385_{\pm 0.035}$	$89.9_{\pm 0.8}$	$1.547_{\pm 0.034}$
	Ours (Algorithm 2, SMLD)	$94.9_{\pm 0.5}$	$1.433_{\pm 0.058}$	$89.9_{\pm 0.7}$	$1.594_{\pm 0.060}$
	FlowCP (LSTM + encoder)	$94.8_{\pm 0.6}$	$1.586_{\pm 0.197}$	$90.0_{\pm 0.9}$	$1.742_{\pm 0.194}$
	FlowCP (LOO bootstrap + encoder)	$94.9_{\pm 0.6}$	$1.432_{\pm 0.080}$	$89.8_{\pm 0.8}$	$1.590_{\pm 0.080}$
	MultiDimSPCI (LSTM)	$94.8_{\pm 0.1}$	$1.369_{\pm 0.230}$	$89.7_{\pm 0.2}$	$1.449_{\pm 0.265}$
	MultiDimSPCI (LOO bootstrap)	$94.8_{\pm 0.1}$	$\mathbf{1.161_{\pm 0.013}}$	$89.8_{\pm 0.2}$	$\mathbf{1.223_{\pm 0.014}}$
	CONTRA (LSTM)	$95.0_{\pm 0.6}$	$1.521_{\pm 0.144}$	$89.9_{\pm 0.8}$	$1.561_{\pm 0.104}$
	CONTRA (LOO bootstrap)	$94.9_{\pm 0.9}$	$1.465_{\pm 0.118}$	$89.9_{\pm 0.9}$	$1.544_{\pm 0.105}$
	Mahalanobis ellipsoid	$94.9_{\pm 0.5}$	$1.371_{\pm 0.026}$	$89.8_{\pm 0.7}$	$1.531_{\pm 0.025}$
1000	Ours (Algorithm 1, OU)	$94.8_{\pm 0.7}$	$\underline{1.131}_{\pm 0.057}$	$89.9_{\pm 0.8}$	$\underline{1.175}_{\pm 0.045}$
	Ours (Algorithm 1, DDPM)	$94.8_{\pm 0.6}$	$1.400_{\pm 0.040}$	$89.7_{\pm 0.8}$	$1.564_{\pm 0.038}$
	Ours (Algorithm 1, SMLD)	$94.8_{\pm 0.6}$	$1.428_{\pm 0.059}$	$89.7_{\pm 0.7}$	$1.590_{\pm 0.055}$
	Ours (Algorithm 2, OU)	$95.0_{\pm 0.6}$	$\mathbf{1.129_{\pm 0.045}}$	$90.0_{\pm 0.7}$	$\mathbf{1.163_{\pm 0.039}}$
	Ours (Algorithm 2, DDPM)	$94.9_{\pm 0.5}$	$1.271_{\pm 0.086}$	$89.8_{\pm 0.8}$	$1.294_{\pm 0.112}$
	Ours (Algorithm 2, SMLD)	$94.9_{\pm 0.5}$	$1.395_{\pm 0.039}$	$89.9_{\pm 0.6}$	$1.557_{\pm 0.042}$
	FlowCP (LSTM + encoder)	$94.9_{\pm 0.7}$	$1.461_{\pm 0.172}$	$89.9_{\pm 0.8}$	$1.616_{\pm 0.170}$
	FlowCP (LOO bootstrap + encoder)	$94.9_{\pm 0.5}$	$1.399_{\pm 0.046}$	$89.8_{\pm 0.8}$	$1.556_{\pm 0.048}$
	MultiDimSPCI (LSTM)	$94.8_{\pm 0.1}$	$1.240_{\pm 0.042}$	$89.8_{\pm 0.2}$	$1.302_{\pm 0.042}$
	MultiDimSPCI (LOO bootstrap)	$94.8_{\pm 0.1}$	$1.162_{\pm 0.012}$	$89.9_{\pm 0.2}$	$1.220_{\pm 0.013}$
	CONTRA (LSTM)	$94.9_{\pm 0.6}$	$1.456_{\pm 0.126}$	$89.9_{\pm 1.0}$	$1.533_{\pm 0.122}$
	CONTRA (LOO bootstrap)	$94.8_{\pm 0.7}$	$1.416_{\pm 0.104}$	$89.9_{\pm 0.7}$	$1.487_{\pm 0.112}$
	Mahalanobis ellipsoid	$94.9_{\pm 0.6}$	$1.346_{\pm 0.023}$	$89.8_{\pm 0.6}$	$1.504_{\pm 0.022}$
2000	Ours (Algorithm 1, OU)	$95.0_{\pm 0.6}$	$\underline{1.082}_{\pm 0.032}$	$90.0_{\pm 0.7}$	$\underline{1.110}_{\pm 0.032}$
	Ours (Algorithm 1, DDPM)	$95.1_{\pm 0.6}$	$1.127_{\pm 0.047}$	$90.0_{\pm 0.8}$	$1.158_{\pm 0.035}$
	Ours (Algorithm 1, SMLD)	$94.9_{\pm 0.6}$	$1.397_{\pm 0.054}$	$90.0_{\pm 0.8}$	$1.557_{\pm 0.060}$
	Ours (Algorithm 2, OU)	$95.0_{\pm 0.5}$	$\mathbf{1.079_{\pm 0.031}}$	$90.0_{\pm 0.7}$	$\mathbf{1.106_{\pm 0.031}}$
	Ours (Algorithm 2, DDPM)	$95.0_{\pm 0.5}$	$1.105_{\pm 0.034}$	$90.0_{\pm 0.7}$	$1.144_{\pm 0.029}$
	Ours (Algorithm 2, SMLD)	$94.8_{\pm 0.5}$	$1.329_{\pm 0.054}$	$89.9_{\pm 0.7}$	$1.424_{\pm 0.120}$
	FlowCP (LSTM + encoder)	$94.9_{\pm 0.5}$	$1.411_{\pm 0.064}$	$89.8_{\pm 0.7}$	$1.566_{\pm 0.063}$
	FlowCP (LOO bootstrap + encoder)	$94.9_{\pm 0.5}$	$1.391_{\pm 0.060}$	$89.9_{\pm 0.8}$	$1.547_{\pm 0.053}$
	MultiDimSPCI (LSTM)	$94.8_{\pm 0.2}$	$1.199_{\pm 0.037}$	$89.8_{\pm 0.2}$	$1.256_{\pm 0.038}$
	MultiDimSPCI (LOO bootstrap)	$94.9_{\pm 0.2}$	$1.162_{\pm 0.014}$	$89.8_{\pm 0.2}$	$1.219_{\pm 0.013}$
	CONTRA (LSTM)	$94.9_{\pm 0.5}$	$1.406_{\pm 0.088}$	$90.0_{\pm 0.7}$	$1.507_{\pm 0.090}$
	CONTRA (LOO bootstrap)	$94.9_{\pm 0.5}$	$1.403_{\pm 0.133}$	$90.0_{\pm 0.6}$	$1.483_{\pm 0.102}$
	Mahalanobis ellipsoid	$94.9_{\pm 0.5}$	$1.333_{\pm 0.016}$	$89.8_{\pm 0.6}$	$1.495_{\pm 0.017}$
4000	Ours (Algorithm 1, OU)	$95.0_{\pm 0.5}$	$\underline{1.069}_{\pm 0.037}$	$90.0_{\pm 0.6}$	$\underline{1.095}_{\pm 0.034}$
	Ours (Algorithm 1, DDPM)	$95.0_{\pm 0.5}$	$1.090_{\pm 0.028}$	$90.0_{\pm 0.7}$	$1.121_{\pm 0.031}$
	Ours (Algorithm 1, SMLD)	$95.0_{\pm 0.5}$	$1.090_{\pm 0.036}$	$90.2_{\pm 0.7}$	$1.131_{\pm 0.046}$
	Ours (Algorithm 2, OU)	$95.0_{\pm 0.6}$	$\mathbf{1.055_{\pm 0.028}}$	$90.0_{\pm 0.8}$	$\mathbf{1.076_{\pm 0.032}}$
	Ours (Algorithm 2, DDPM)	$95.1_{\pm 0.6}$	$1.077_{\pm 0.028}$	$90.1_{\pm 0.8}$	$1.105_{\pm 0.033}$
	Ours (Algorithm 2, SMLD)	$95.0_{\pm 0.7}$	$1.084_{\pm 0.036}$	$90.0_{\pm 0.8}$	$1.115_{\pm 0.040}$
	FlowCP (LSTM + encoder)	$94.9_{\pm 0.6}$	$1.398_{\pm 0.045}$	$89.7_{\pm 0.6}$	$1.551_{\pm 0.042}$
	FlowCP (LOO bootstrap + encoder)	$94.9_{\pm 0.5}$	$1.438_{\pm 0.147}$	$89.7_{\pm 0.6}$	$1.570_{\pm 0.117}$
	MultiDimSPCI (LSTM)	$94.8_{\pm 0.2}$	$1.182_{\pm 0.018}$	$89.8_{\pm 0.2}$	$1.242_{\pm 0.019}$
	MultiDimSPCI (LOO bootstrap)	$94.9_{\pm 0.2}$	$1.157_{\pm 0.012}$	$89.9_{\pm 0.2}$	$1.216_{\pm 0.013}$
	CONTRA (LSTM)	$94.9_{\pm 0.6}$	$1.405_{\pm 0.137}$	$89.8_{\pm 0.8}$	$1.483_{\pm 0.127}$
	CONTRA (LOO bootstrap)	$95.0_{\pm 0.5}$	$1.398_{\pm 0.126}$	$89.7_{\pm 0.8}$	$1.479_{\pm 0.121}$
	Mahalanobis ellipsoid	$94.9_{\pm 0.6}$	$1.330_{\pm 0.017}$	$89.7_{\pm 0.6}$	$1.488_{\pm 0.015}$
8000	Ours (Algorithm 1, OU)	$94.9_{\pm 0.5}$	$\underline{1.058}_{\pm 0.027}$	$89.9_{\pm 0.6}$	$\underline{1.074}_{\pm 0.025}$
	Ours (Algorithm 1, DDPM)	$94.9_{\pm 0.6}$	$1.073_{\pm 0.033}$	$89.8_{\pm 0.9}$	$1.099_{\pm 0.036}$
	Ours (Algorithm 1, SMLD)	$94.9_{\pm 0.5}$	$1.062_{\pm 0.029}$	$89.9_{\pm 0.7}$	$1.082_{\pm 0.030}$
	Ours (Algorithm 2, OU)	$95.0_{\pm 0.5}$	$\mathbf{1.047_{\pm 0.032}}$	$89.9_{\pm 0.7}$	$\mathbf{1.059_{\pm 0.031}}$
	Ours (Algorithm 2, DDPM)	$95.0_{\pm 0.5}$	$1.067_{\pm 0.029}$	$89.8_{\pm 0.7}$	$1.083_{\pm 0.029}$
	Ours (Algorithm 2, SMLD)	$94.9_{\pm 0.5}$	$1.071_{\pm 0.034}$	$89.8_{\pm 0.7}$	$1.088_{\pm 0.037}$
	FlowCP (LSTM + encoder)	$94.9_{\pm 0.5}$	$1.470_{\pm 0.286}$	$89.8_{\pm 0.6}$	$1.613_{\pm 0.250}$
	FlowCP (LOO bootstrap + encoder)	$95.0_{\pm 0.5}$	$1.382_{\pm 0.052}$	$89.8_{\pm 0.6}$	$1.529_{\pm 0.048}$
	MultiDimSPCI (LSTM)	$94.9_{\pm 0.2}$	$1.180_{\pm 0.016}$	$89.9_{\pm 0.3}$	$1.240_{\pm 0.016}$
	MultiDimSPCI (LOO bootstrap)	$95.0_{\pm 0.3}$	$1.159_{\pm 0.012}$	$89.9_{\pm 0.4}$	$1.218_{\pm 0.011}$
	CONTRA (LSTM)	$94.9_{\pm 0.4}$	$1.394_{\pm 0.139}$	$90.0_{\pm 0.5}$	$1.469_{\pm 0.125}$
	CONTRA (LOO bootstrap)	$94.8_{\pm 0.4}$	$1.369_{\pm 0.111}$	$89.8_{\pm 0.5}$	$1.442_{\pm 0.112}$
	Mahalanobis ellipsoid	$95.0_{\pm 0.6}$	$1.328_{\pm 0.018}$	$89.7_{\pm 0.6}$	$1.484_{\pm 0.015}$

Table 6: VAR(1) prediction regions under spherical Gaussian-mixture innovations with 8.3σ separation. CVR (%) denotes empirical coverage, and volume is normalized by the oracle highest-density region. Results are reported as mean $\pm$ one standard deviation over 30 seeds. **Bold** and underlining denote the smallest and second-smallest mean volumes, respectively, for each n and α.

		$\alpha = 0.05$		$\alpha = 0.10$	
n	Method	CVR (%)	Volume Ratio	CVR (%)	Volume Ratio
500	Ours (Algorithm 1, OU)	$94.9_{\pm 0.7}$	$3.177_{\pm 0.165}$	$89.9_{\pm 0.8}$	$3.769_{\pm 0.197}$
	Ours (Algorithm 1, DDPM)	$95.0_{\pm 0.7}$	$3.180_{\pm 0.172}$	$89.9_{\pm 0.9}$	$3.730_{\pm 0.163}$
	Ours (Algorithm 1, SMLD)	$95.1_{\pm 0.5}$	$3.293_{\pm 0.204}$	$90.1_{\pm 0.8}$	$3.871_{\pm 0.225}$
	Ours (Algorithm 2, OU)	$95.1_{\pm 0.5}$	$\underline{2.018}_{\pm 0.218}$	$90.0_{\pm 0.7}$	$\underline{2.131}_{\pm 0.249}$
	Ours (Algorithm 2, DDPM)	$95.1_{\pm 0.5}$	$3.046_{\pm 0.123}$	$89.9_{\pm 0.8}$	$3.643_{\pm 0.148}$
	Ours (Algorithm 2, SMLD)	$94.9_{\pm 0.4}$	$3.130_{\pm 0.218}$	$89.9_{\pm 0.6}$	$3.720_{\pm 0.238}$
	FlowCP (LSTM + encoder)	$95.0_{\pm 0.7}$	$3.482_{\pm 0.569}$	$90.1_{\pm 0.8}$	$4.054_{\pm 0.578}$
	FlowCP (LOO bootstrap + encoder)	$95.0_{\pm 0.5}$	$3.126_{\pm 0.176}$	$90.1_{\pm 0.7}$	$3.719_{\pm 0.200}$
	MultiDimSPCI (LSTM)	$95.0_{\pm 0.4}$	$2.992_{\pm 1.146}$	$90.1_{\pm 0.4}$	$3.327_{\pm 1.392}$
	MultiDimSPCI (LOO bootstrap)	$94.8_{\pm 0.1}$	$\mathbf{1.856_{\pm 0.028}}$	$89.8_{\pm 0.2}$	$\mathbf{2.023_{\pm 0.033}}$
	CONTRA (LSTM)	$95.1_{\pm 0.6}$	$2.966_{\pm 1.119}$	$90.1_{\pm 0.9}$	$2.544_{\pm 0.630}$
	CONTRA (LOO bootstrap)	$95.1_{\pm 0.7}$	$2.912_{\pm 1.180}$	$90.1_{\pm 0.8}$	$2.656_{\pm 0.679}$
	Mahalanobis ellipsoid	$95.1_{\pm 0.5}$	$2.967_{\pm 0.085}$	$89.9_{\pm 0.8}$	$3.533_{\pm 0.093}$
1000	Ours (Algorithm 1, OU)	$95.0_{\pm 0.5}$	$\mathbf{1.392_{\pm 0.122}}$	$89.9_{\pm 0.7}$	$\mathbf{1.329_{\pm 0.125}}$
	Ours (Algorithm 1, DDPM)	$94.9_{\pm 0.6}$	$2.903_{\pm 0.260}$	$89.9_{\pm 0.7}$	$3.325_{\pm 0.417}$
	Ours (Algorithm 1, SMLD)	$95.0_{\pm 0.6}$	$3.119_{\pm 0.160}$	$90.0_{\pm 0.7}$	$3.679_{\pm 0.158}$
	Ours (Algorithm 2, OU)	$95.0_{\pm 0.5}$	$\underline{1.430}_{\pm 0.093}$	$90.1_{\pm 0.7}$	$\underline{1.369}_{\pm 0.126}$
	Ours (Algorithm 2, DDPM)	$95.0_{\pm 0.4}$	$1.811_{\pm 0.095}$	$90.0_{\pm 0.5}$	$1.898_{\pm 0.126}$
	Ours (Algorithm 2, SMLD)	$95.1_{\pm 0.5}$	$3.015_{\pm 0.111}$	$89.9_{\pm 0.7}$	$3.591_{\pm 0.124}$
	FlowCP (LSTM + encoder)	$95.0_{\pm 0.6}$	$3.158_{\pm 0.201}$	$90.1_{\pm 0.7}$	$3.739_{\pm 0.206}$
	FlowCP (LOO bootstrap + encoder)	$95.1_{\pm 0.6}$	$3.030_{\pm 0.205}$	$90.0_{\pm 0.7}$	$3.611_{\pm 0.231}$
	MultiDimSPCI (LSTM)	$94.8_{\pm 0.1}$	$2.162_{\pm 0.210}$	$89.8_{\pm 0.2}$	$2.358_{\pm 0.238}$
	MultiDimSPCI (LOO bootstrap)	$94.8_{\pm 0.1}$	$1.850_{\pm 0.029}$	$89.8_{\pm 0.2}$	$2.017_{\pm 0.028}$
	CONTRA (LSTM)	$95.1_{\pm 0.8}$	$2.752_{\pm 0.816}$	$90.3_{\pm 0.9}$	$2.501_{\pm 0.712}$
	CONTRA (LOO bootstrap)	$95.1_{\pm 0.7}$	$2.764_{\pm 0.989}$	$90.2_{\pm 0.9}$	$2.492_{\pm 0.794}$
	Mahalanobis ellipsoid	$95.0_{\pm 0.4}$	$2.881_{\pm 0.040}$	$89.9_{\pm 0.7}$	$3.446_{\pm 0.044}$
2000	Ours (Algorithm 1, OU)	$95.0_{\pm 0.5}$	$\mathbf{1.202_{\pm 0.096}}$	$90.1_{\pm 0.8}$	$\mathbf{1.144_{\pm 0.068}}$
	Ours (Algorithm 1, DDPM)	$95.0_{\pm 0.5}$	$1.513_{\pm 0.105}$	$90.0_{\pm 0.6}$	$1.459_{\pm 0.128}$
	Ours (Algorithm 1, SMLD)	$95.0_{\pm 0.4}$	$2.980_{\pm 0.229}$	$90.0_{\pm 0.8}$	$3.540_{\pm 0.303}$
	Ours (Algorithm 2, OU)	$95.0_{\pm 0.4}$	$\underline{1.257}_{\pm 0.065}$	$90.0_{\pm 0.7}$	$\underline{1.191}_{\pm 0.054}$
	Ours (Algorithm 2, DDPM)	$95.0_{\pm 0.5}$	$1.478_{\pm 0.085}$	$90.0_{\pm 0.7}$	$1.402_{\pm 0.107}$
	Ours (Algorithm 2, SMLD)	$95.0_{\pm 0.5}$	$2.133_{\pm 0.385}$	$89.9_{\pm 0.7}$	$2.291_{\pm 0.479}$
	FlowCP (LSTM + encoder)	$95.0_{\pm 0.5}$	$2.986_{\pm 0.105}$	$89.8_{\pm 0.8}$	$3.567_{\pm 0.120}$
	FlowCP (LOO bootstrap + encoder)	$95.0_{\pm 0.5}$	$2.994_{\pm 0.133}$	$89.8_{\pm 0.6}$	$3.556_{\pm 0.137}$
	MultiDimSPCI (LSTM)	$94.8_{\pm 0.1}$	$1.990_{\pm 0.106}$	$89.8_{\pm 0.2}$	$2.166_{\pm 0.119}$
	MultiDimSPCI (LOO bootstrap)	$94.9_{\pm 0.1}$	$1.844_{\pm 0.023}$	$89.9_{\pm 0.2}$	$2.010_{\pm 0.026}$
	CONTRA (LSTM)	$95.1_{\pm 0.7}$	$2.811_{\pm 1.089}$	$90.1_{\pm 0.9}$	$2.665_{\pm 0.959}$
	CONTRA (LOO bootstrap)	$95.2_{\pm 0.7}$	$2.505_{\pm 0.993}$	$90.2_{\pm 0.9}$	$2.256_{\pm 0.797}$
	Mahalanobis ellipsoid	$94.9_{\pm 0.5}$	$2.830_{\pm 0.035}$	$89.9_{\pm 0.7}$	$3.397_{\pm 0.040}$
4000	Ours (Algorithm 1, OU)	$95.1_{\pm 0.6}$	$\underline{1.173}_{\pm 0.077}$	$90.1_{\pm 0.8}$	$\mathbf{1.124_{\pm 0.064}}$
	Ours (Algorithm 1, DDPM)	$95.0_{\pm 0.6}$	$1.342_{\pm 0.123}$	$90.1_{\pm 0.7}$	$1.267_{\pm 0.124}$
	Ours (Algorithm 1, SMLD)	$95.2_{\pm 0.5}$	$1.261_{\pm 0.114}$	$90.1_{\pm 0.7}$	$1.189_{\pm 0.091}$
	Ours (Algorithm 2, OU)	$94.9_{\pm 0.4}$	$\mathbf{1.165_{\pm 0.077}}$	$90.1_{\pm 0.7}$	$\underline{1.124}_{\pm 0.057}$
	Ours (Algorithm 2, DDPM)	$95.1_{\pm 0.4}$	$1.341_{\pm 0.105}$	$90.0_{\pm 0.8}$	$1.238_{\pm 0.107}$
	Ours (Algorithm 2, SMLD)	$95.1_{\pm 0.5}$	$1.263_{\pm 0.119}$	$90.0_{\pm 0.8}$	$1.193_{\pm 0.132}$
	FlowCP (LSTM + encoder)	$95.0_{\pm 0.5}$	$3.010_{\pm 0.118}$	$90.0_{\pm 0.7}$	$3.580_{\pm 0.123}$
	FlowCP (LOO bootstrap + encoder)	$94.9_{\pm 0.4}$	$3.047_{\pm 0.241}$	$89.9_{\pm 0.7}$	$3.591_{\pm 0.212}$
	MultiDimSPCI (LSTM)	$94.9_{\pm 0.1}$	$1.909_{\pm 0.035}$	$89.8_{\pm 0.2}$	$2.071_{\pm 0.043}$
	MultiDimSPCI (LOO bootstrap)	$94.9_{\pm 0.1}$	$1.840_{\pm 0.022}$	$90.0_{\pm 0.2}$	$1.997_{\pm 0.022}$
	CONTRA (LSTM)	$95.0_{\pm 0.5}$	$2.504_{\pm 0.925}$	$90.0_{\pm 0.7}$	$2.259_{\pm 0.721}$
	CONTRA (LOO bootstrap)	$95.1_{\pm 0.6}$	$2.577_{\pm 0.797}$	$90.1_{\pm 0.7}$	$2.403_{\pm 0.729}$
	Mahalanobis ellipsoid	$94.9_{\pm 0.5}$	$2.813_{\pm 0.027}$	$89.8_{\pm 0.7}$	$3.369_{\pm 0.034}$
8000	Ours (Algorithm 1, OU)	$95.0_{\pm 0.5}$	$\underline{1.121}_{\pm 0.064}$	$90.1_{\pm 0.7}$	$\underline{1.099}_{\pm 0.056}$
	Ours (Algorithm 1, DDPM)	$95.0_{\pm 0.6}$	$1.319_{\pm 0.123}$	$89.9_{\pm 0.8}$	$1.231_{\pm 0.123}$
	Ours (Algorithm 1, SMLD)	$95.0_{\pm 0.6}$	$1.170_{\pm 0.098}$	$90.1_{\pm 0.9}$	$1.134_{\pm 0.078}$
	Ours (Algorithm 2, OU)	$95.0_{\pm 0.5}$	$\mathbf{1.105_{\pm 0.067}}$	$90.1_{\pm 0.8}$	$\mathbf{1.086_{\pm 0.055}}$
	Ours (Algorithm 2, DDPM)	$95.0_{\pm 0.5}$	$1.217_{\pm 0.083}$	$90.0_{\pm 0.7}$	$1.155_{\pm 0.075}$
	Ours (Algorithm 2, SMLD)	$95.0_{\pm 0.5}$	$1.157_{\pm 0.085}$	$90.1_{\pm 0.7}$	$1.128_{\pm 0.074}$
	FlowCP (LSTM + encoder)	$94.8_{\pm 0.5}$	$3.297_{\pm 1.259}$	$90.1_{\pm 0.7}$	$3.817_{\pm 1.002}$
	FlowCP (LOO bootstrap + encoder)	$95.0_{\pm 0.5}$	$2.883_{\pm 0.107}$	$89.9_{\pm 0.7}$	$3.425_{\pm 0.111}$
	MultiDimSPCI (LSTM)	$94.9_{\pm 0.2}$	$1.906_{\pm 0.046}$	$89.8_{\pm 0.2}$	$2.077_{\pm 0.052}$
	MultiDimSPCI (LOO bootstrap)	$94.9_{\pm 0.2}$	$1.833_{\pm 0.023}$	$89.9_{\pm 0.3}$	$1.996_{\pm 0.021}$
	CONTRA (LSTM)	$95.0_{\pm 0.5}$	$2.363_{\pm 0.659}$	$90.0_{\pm 0.8}$	$2.183_{\pm 0.516}$
	CONTRA (LOO bootstrap)	$95.0_{\pm 0.5}$	$2.331_{\pm 0.719}$	$89.9_{\pm 0.7}$	$2.187_{\pm 0.700}$
	Mahalanobis ellipsoid	$95.0_{\pm 0.5}$	$2.802_{\pm 0.030}$	$89.9_{\pm 0.7}$	$3.369_{\pm 0.039}$

density g, the volume of a prediction region satisfies

$$V(x) = \int_{\mathbb{R}^{d_y}} 1\left\{y \in \widehat{\mathcal{C}}_{1-\alpha}(x)\right\}\, dy = \mathbb{E}_{Y\sim g}\left[\frac{1\left\{Y \in \widehat{\mathcal{C}}_{1-\alpha}(x)\right\}}{g(Y)}\right].$$

We construct g exclusively from the calibration responses, or from calibration residuals for residual-based methods, using the defensive mixture

$$g(y) = 0.8\,\frac{1}{K}\sum_{k=1}^{K}\phi_{d_y}\left(y; z_k, h^2\widehat{\Sigma}_z\right) + 0.2\,t_{3,d_y}\left(y; \bar{z}, 3^2\widehat{\Sigma}_z\right),$$

where $K = \min(256, n_{\text{cal}})$, $h = n_{\text{cal}}^{-1/(d_y+4)}$ is Scott's bandwidth, and the multivariate Student-t component provides defensive tail coverage. For each independent replicate r, we draw N_{d_y} samples $Y_{ri} \sim g$ and compute

$$\widehat{V}_r = \frac{1}{N_{d_y}}\sum_{i=1}^{N_{d_y}}\frac{1\left\{Y_{ri} \in \widehat{\mathcal{C}}_{1-\alpha}(x_{j(i)})\right\}}{g(Y_{ri})}, \qquad \widehat{V} = \frac{1}{R}\sum_{r=1}^{R}\widehat{V}_r,$$

where $j(i)$ cycles uniformly over the evaluated test contexts. In paper mode, we use $N_{d_y} = 4096$, 8192, and 16384 samples for $d_y = 2$, 4, and 8, respectively, with $R = 8$ independent replicates. Numerical integration uncertainty is quantified by

$$\widehat{\text{SE}}_{\text{IS}} = \frac{\text{sd}(\widehat{V}_1, \ldots, \widehat{V}_R)}{\sqrt{R}}.$$

We additionally monitor the effective sample size $\text{ESS} = (\sum_i w_i)^2/\sum_i w_i^2$ and the maximum normalized weight $\rho_{\max} = \max_i w_i/\sum_i w_i$, where $w_i = 1\{Y_i \in \widehat{\mathcal{C}}\}/g(Y_i)$. An estimate is flagged as numerically unreliable when $\text{ESS} < 200$ or $\rho_{\max} > 0.05$. Volumes are reported in the original outcome units, together with the equivalent side length $\widehat{V}^{1/d_y}$ and normalized log-volume $\log(\widehat{V})/d_y$. Final results are summarized as the mean and sample standard deviation across the five location folds; the importance-sampling standard error measures numerical integration uncertainty and is recorded separately.

APPENDIX C ABLATION STUDIES

APPENDIX C.1 DETAILS OF ABLATION STUDIES IN REAL DATA

We focus on the strongest multimodal 8.3σ spherical Gaussian mixture experiment with sample size $n = 2000$ under the OU diffusion backbone. We keep the trained diffusion and the number of MC budget $|\mathcal{T}|R$.

Now we can replace these three components. First, replace the stratified time samples with $v_k \overset{\text{iid}}{\sim} \text{Unif}(0,1)$. Second, replace the antithetic perturbations with $\epsilon_{k,1}, \ldots, \epsilon_{k,R} \overset{\text{iid}}{\sim} \mathcal{N}(0, I_{d_y})$. And we can use an independently generated perturbation bank $w_i = \{v_{i,k}, \varepsilon_{i,k,r}\}_{k,r}$, rather than a fixed random number bank across calibration observations and test candidates.

APPENDIX C.2 ABLATION STUDIES IN SYNTHETIC DATA

We conduct an additional ablation study to assess these three components mentioned in Section 5. We use the same OU diffusion backbone and total number of denoiser evaluations. Table 8 reports the average empirical coverage, prediction region volume, and Monte Carlo (MC) Score SD. The volume and MC Score SD can increase by 1.1–1.5% and 36–39% without stratification, respectively. The stratified time sampling can stabilize the score across the MC banks. Moreover, the volume can increase by 3.4–18.1% without antithetic perturbations or CRN banks.

Table 7: Ablation study of the finite-budget score construction on the Traffic and Solar datasets. All variants use the OU diffusion backbone and the same number of denoiser evaluations. CVR denotes empirical coverage, and MC Score SD denotes the variability of the nonconformity score across independently generated Monte Carlo banks. Results are reported as mean ± one standard deviation over five folds.

				$\alpha = 0.05$		$\alpha = 0.10$		
Dataset	d_y	Algorithm	Variant	CVR (%)	Volume	CVR (%)	Volume	MC Score SD
Traffic	2	Algorithm 1	Full	$93.99_{\pm 5.26}$	$0.491_{\pm 0.334}$	$88.28_{\pm 8.27}$	$0.295_{\pm 0.187}$	$0.060_{\pm 0.012}$
			w/o stratification	$93.83_{\pm 5.41}$	$0.496_{\pm 0.339}$	$88.22_{\pm 8.36}$	$0.298_{\pm 0.186}$	$0.087_{\pm 0.021}$
			w/o antithetic	$93.97_{\pm 5.23}$	$0.504_{\pm 0.347}$	$88.31_{\pm 8.35}$	$0.301_{\pm 0.190}$	$0.059_{\pm 0.015}$
			w/o CRN	$94.04_{\pm 5.11}$	$0.502_{\pm 0.335}$	$88.65_{\pm 7.98}$	$0.305_{\pm 0.189}$	$0.056_{\pm 0.014}$
		Algorithm 2	Full	$97.01_{\pm 1.51}$	$0.895_{\pm 0.390}$	$92.92_{\pm 3.64}$	$0.542_{\pm 0.215}$	$0.133_{\pm 0.006}$
			w/o stratification	$97.00_{\pm 1.68}$	$0.915_{\pm 0.387}$	$92.92_{\pm 3.55}$	$0.548_{\pm 0.217}$	$0.163_{\pm 0.003}$
			w/o antithetic	$96.90_{\pm 1.66}$	$0.946_{\pm 0.406}$	$92.91_{\pm 3.17}$	$0.570_{\pm 0.220}$	$0.112_{\pm 0.004}$
			w/o CRN	$96.82_{\pm 1.64}$	$1.153_{\pm 0.470}$	$92.97_{\pm 2.79}$	$0.702_{\pm 0.263}$	$0.140_{\pm 0.003}$
	4	Algorithm 1	Full	$93.06_{\pm 9.69}$	$0.483_{\pm 0.320}$	$87.56_{\pm 11.07}$	$0.188_{\pm 0.117}$	$0.082_{\pm 0.010}$
			w/o stratification	$93.01_{\pm 9.44}$	$0.503_{\pm 0.322}$	$87.44_{\pm 11.25}$	$0.195_{\pm 0.123}$	$0.286_{\pm 0.115}$
			w/o antithetic	$93.12_{\pm 9.25}$	$0.505_{\pm 0.334}$	$87.51_{\pm 11.22}$	$0.194_{\pm 0.122}$	$0.090_{\pm 0.022}$
			w/o CRN	$93.08_{\pm 9.45}$	$0.500_{\pm 0.333}$	$87.46_{\pm 11.21}$	$0.201_{\pm 0.129}$	$0.097_{\pm 0.006}$
		Algorithm 2	Full	$96.95_{\pm 1.63}$	$1.220_{\pm 0.735}$	$93.45_{\pm 2.47}$	$0.506_{\pm 0.291}$	$0.186_{\pm 0.005}$
			w/o stratification	$96.93_{\pm 1.63}$	$1.280_{\pm 0.768}$	$93.42_{\pm 2.52}$	$0.520_{\pm 0.302}$	$0.368_{\pm 0.008}$
			w/o antithetic	$96.99_{\pm 1.70}$	$1.301_{\pm 0.788}$	$93.34_{\pm 2.44}$	$0.545_{\pm 0.314}$	$0.146_{\pm 0.006}$
			w/o CRN	$96.86_{\pm 1.52}$	$1.844_{\pm 1.166}$	$92.98_{\pm 2.43}$	$0.809_{\pm 0.484}$	$0.203_{\pm 0.004}$
Solar	2	Algorithm 1	Full	$94.95_{\pm 0.51}$	$0.320_{\pm 0.081}$	$88.60_{\pm 1.47}$	$0.118_{\pm 0.016}$	$0.046_{\pm 0.004}$
			w/o stratification	$94.95_{\pm 0.41}$	$0.327_{\pm 0.078}$	$88.55_{\pm 1.47}$	$0.123_{\pm 0.017}$	$0.070_{\pm 0.003}$
			w/o antithetic	$94.93_{\pm 0.36}$	$0.324_{\pm 0.076}$	$88.62_{\pm 1.49}$	$0.121_{\pm 0.014}$	$0.039_{\pm 0.002}$
			w/o CRN	$94.78_{\pm 0.28}$	$0.359_{\pm 0.084}$	$88.66_{\pm 1.26}$	$0.145_{\pm 0.020}$	$0.038_{\pm 0.002}$
		Algorithm 2	Full	$94.76_{\pm 0.83}$	$0.881_{\pm 0.119}$	$88.45_{\pm 2.19}$	$0.330_{\pm 0.058}$	$0.114_{\pm 0.005}$
			w/o stratification	$94.78_{\pm 0.79}$	$1.054_{\pm 0.234}$	$88.44_{\pm 2.06}$	$0.349_{\pm 0.061}$	$0.151_{\pm 0.009}$
			w/o antithetic	$94.87_{\pm 0.79}$	$0.996_{\pm 0.135}$	$88.52_{\pm 2.13}$	$0.356_{\pm 0.052}$	$0.091_{\pm 0.002}$
			w/o CRN	$94.78_{\pm 0.67}$	$1.332_{\pm 0.216}$	$88.56_{\pm 1.78}$	$0.456_{\pm 0.072}$	$0.100_{\pm 0.002}$
	4	Algorithm 1	Full	$95.01_{\pm 0.70}$	$0.325_{\pm 0.154}$	$87.37_{\pm 1.41}$	$0.037_{\pm 0.020}$	$0.048_{\pm 0.002}$
			w/o stratification	$94.81_{\pm 0.52}$	$0.349_{\pm 0.152}$	$87.77_{\pm 1.31}$	$0.045_{\pm 0.025}$	$0.128_{\pm 0.002}$
			w/o antithetic	$94.94_{\pm 0.51}$	$0.349_{\pm 0.162}$	$87.55_{\pm 1.24}$	$0.039_{\pm 0.020}$	$0.044_{\pm 0.002}$
			w/o CRN	$94.60_{\pm 0.51}$	$0.449_{\pm 0.163}$	$87.87_{\pm 1.14}$	$0.070_{\pm 0.032}$	$0.056_{\pm 0.003}$
		Algorithm 2	Full	$95.20_{\pm 0.58}$	$1.002_{\pm 0.441}$	$88.84_{\pm 1.62}$	$0.164_{\pm 0.064}$	$0.127_{\pm 0.002}$
			w/o stratification	$95.33_{\pm 0.62}$	$1.138_{\pm 0.480}$	$88.87_{\pm 1.45}$	$0.174_{\pm 0.070}$	$0.252_{\pm 0.008}$
			w/o antithetic	$95.20_{\pm 0.69}$	$1.142_{\pm 0.479}$	$88.94_{\pm 1.52}$	$0.188_{\pm 0.073}$	$0.098_{\pm 0.002}$
			w/o CRN	$95.01_{\pm 0.60}$	$2.322_{\pm 0.923}$	$89.04_{\pm 1.33}$	$0.354_{\pm 0.156}$	$0.133_{\pm 0.002}$

Table 8: Ablation study of the finite-budget score construction under the 8.3σ multimodal VAR(1) setting with $n = 2000$. All variants use the OU diffusion backbone and the same number of denoiser evaluations. MC Score SD measures the variability of the nonconformity score across independently generated Monte Carlo banks. Results are reported as mean ± one standard deviation over 30 seeds.

		$\alpha = 0.05$		$\alpha = 0.10$		
Algorithm	Variant	CVR (%)	Volume Ratio	CVR (%)	Volume Ratio	MC Score SD
Algorithm 1	Full	$95.04_{\pm 0.52}$	$1.220_{\pm 0.071}$	$90.12_{\pm 0.58}$	$1.164_{\pm 0.051}$	$0.107_{\pm 0.014}$
	w/o stratification	$95.03_{\pm 0.50}$	$1.236_{\pm 0.067}$	$90.11_{\pm 0.59}$	$1.181_{\pm 0.050}$	$0.149_{\pm 0.015}$
	w/o antithetic	$95.01_{\pm 0.48}$	$1.267_{\pm 0.067}$	$90.11_{\pm 0.56}$	$1.212_{\pm 0.049}$	$0.094_{\pm 0.008}$
	w/o CRN	$95.04_{\pm 0.39}$	$1.395_{\pm 0.051}$	$90.11_{\pm 0.50}$	$1.370_{\pm 0.046}$	$0.105_{\pm 0.002}$
Algorithm 2	Full	$95.00_{\pm 0.46}$	$1.274_{\pm 0.060}$	$89.99_{\pm 0.69}$	$1.207_{\pm 0.056}$	$0.114_{\pm 0.015}$
	w/o stratification	$95.00_{\pm 0.47}$	$1.288_{\pm 0.061}$	$89.98_{\pm 0.70}$	$1.224_{\pm 0.060}$	$0.155_{\pm 0.017}$
	w/o antithetic	$95.00_{\pm 0.42}$	$1.318_{\pm 0.066}$	$89.97_{\pm 0.64}$	$1.256_{\pm 0.062}$	$0.099_{\pm 0.009}$
	w/o CRN	$95.02_{\pm 0.38}$	$1.445_{\pm 0.056}$	$90.02_{\pm 0.56}$	$1.426_{\pm 0.061}$	$0.111_{\pm 0.003}$

APPENDIX D MONTE CARLO BUDGET

APPENDIX D.1 FIXED-BUDGET SCORE APPROXIMATION ON SYNTHETIC DATA

This experiment studies how a fixed computational budget is affected by using more diffusion-time levels and using more Monte Carlo samples at each level. We choose different $|\mathcal{T}| \in \{4, 8, 16, 32, 64, 128\}$ with corresponding R such that MC budget $|\mathcal{T}|R = 256$ is fixed.

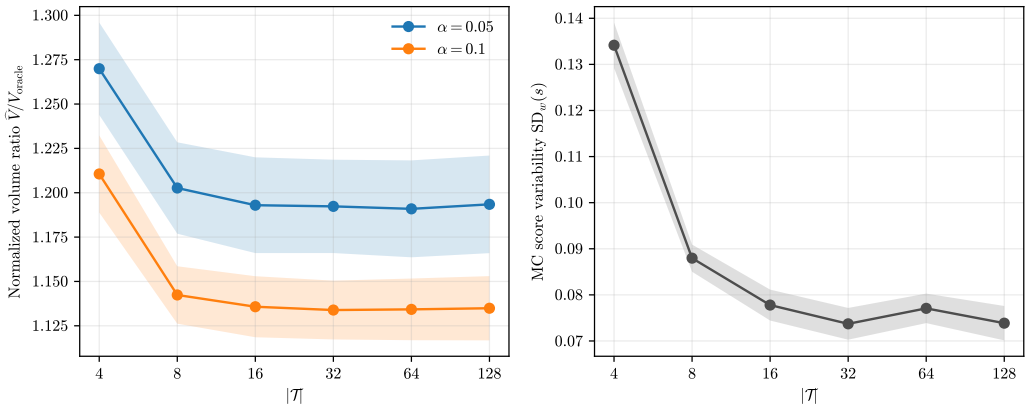


Figure 3: Effect of the number of diffusion-time levels $|\mathcal{T}|$ on prediction-region efficiency and Monte Carlo stability. The left panel reports the volume ratio $\widehat{V}/V_{\text{oracle}}$ for $\alpha \in \{0.05, 0.10\}$, while the right panel reports the variability of the Monte Carlo score across independently generated sampling banks. Points denote mean values and shaded bands indicate $\pm$ one standard deviation.

Figure 3 shows that using smaller diffusion-time levels $|\mathcal{T}|$ leads to larger prediction regions and less stable Monte Carlo scores. Both metrics decrease substantially as $|\mathcal{T}|$ increases from 4 to 16, while they change little after 16. It also suggests the score approximation can benefit most when $|\mathcal{T}| \approx R$.

APPENDIX D.2 FIXED-BUDGET SCORE APPROXIMATION ON REAL DATA

We conduct the same experiment on the Traffic and Solar datasets with $d_y \in \{2, 4\}$ under different $|\mathcal{T}| \in \{4, 8, 16, 32, 64\}$ with corresponding R such that the MC budget $|\mathcal{T}|R = 256$ is fixed. Since the prediction region volumes are not normalized across datasets, we report the raw volume in the upper panels.

Figure 4 shows a pattern similar to that observed in the synthetic experiments. The consistency across synthetic and real-world data supports using $|\mathcal{T}| = 16$ as an optimal setup under another trade-off between the diffusion-time levels and MC samples.

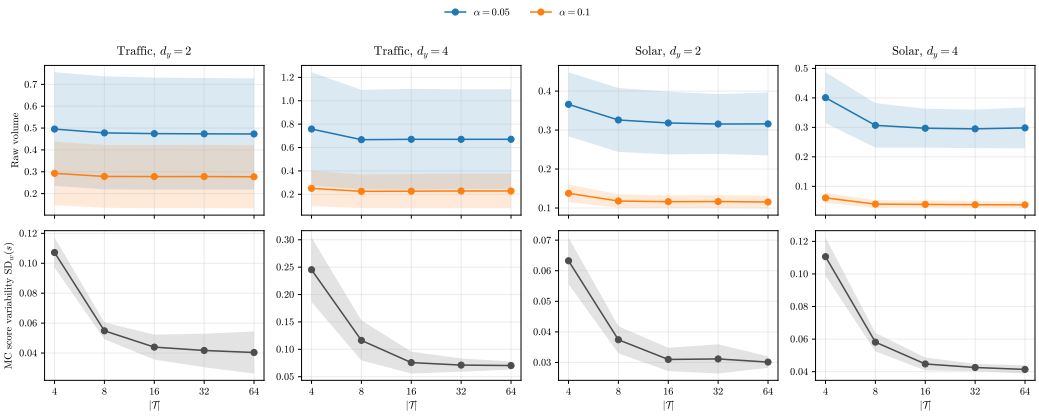


Figure 4: Effect of the number of diffusion-time levels $|\mathcal{T}|$ on prediction-region efficiency and Monte Carlo stability in the real-world datasets under $d_y \in \{2, 4\}$. Upper panels report the volume for $\alpha \in \{0.05, 0.10\}$, while the bottom panels report the variability of the Monte Carlo score across independently generated sampling banks. Points denote mean values and shaded bands indicate $\pm$ one standard deviation.

APPENDIX E TEMPORAL DEPENDENCE SENSITIVITY

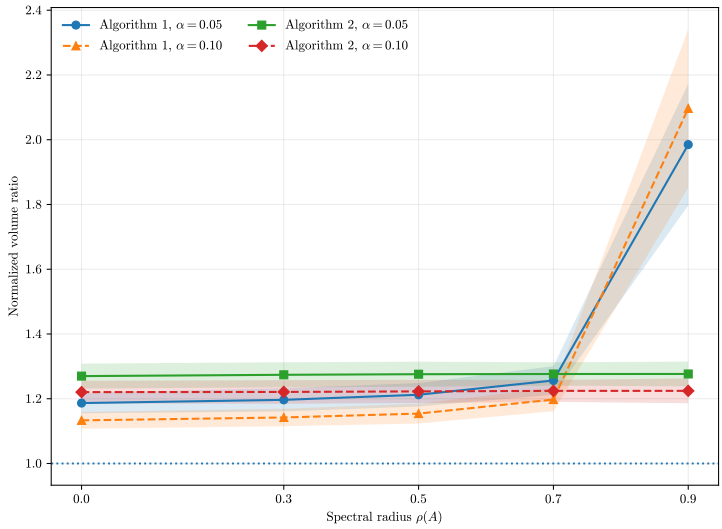


Figure 5: Normalized prediction region volume ratio for Algorithm 1 and 2 with OU diffusion backbone at $\alpha \in \{0.05, 0.10\}$ under different spectral radius $\rho(A)$ in the VAR(1) processes. Markers denote means and shaded bands indicate $\pm$ one standard deviation.

In this section, we conduct numerical experiments to study how the dependence strength affects the method. To preserve the structure of the transition matrix, define

$$A = \frac{\rho}{\rho(A_0)} A_0,$$

where $A_0 = \begin{pmatrix} 0.5 & -0.3 \\ 0.2 & 0.4 \end{pmatrix}$. Then we choose the spectral radius $\rho(A) \in \{0, 0.3, 0.5, 0.7, 0.9\}$. Especially, $\rho(A) = 0$ means i.i.d case. All values satisfy $\rho < 1$, so the resulting VAR process remains stationary.

Figure 5 shows that Algorithm 1 is relatively stable to temporal dependence under moderate spectral radius $\rho(A) \leq 0.7$. The result in Algorithm 1 is also consistent with the increasing dependence penalty account for the finite-sample β-mixing analysis. In contrast, Algorithm 2 is more robust to increasingly strong temporal dependence in time series data.