SHIFT: INPUT-CONDITIONED STOCHASTIC TEMPORAL REPARAMETERIZATION FOR NEURAL OPERATOR FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Time-series forecasting becomes challenging when observations are irregularly sampled, partially missing, asynchronously recorded, or unevenly informative in physical time. Most temporal neural operators operate on a fixed temporal coordinate system, leaving the allocation of latent temporal resolution deterministic. We introduce SHIFT, a neural-operator forecasting architecture that represents this allocation through an input-conditioned stochastic monotone reparameterization of physical time. Given observed values, masks, timestamps, and time gaps, SHIFT samples positive interval rates whose normalized cumulative sums define a strictly monotone mapping from physical time to a normalized latent-time domain. This stochastic temporal mapping resamples the encoded history onto a uniform latent-time grid, where adaptively fused multi-scale spectral–temporal branches process the resulting representation. Multiple sampled temporal reparameterizations define a finite Monte Carlo predictive mixture, allowing predictive uncertainty to be decomposed into conditional value uncertainty and temporal-geometry-induced uncertainty. The latter measures forecast sensitivity to alternative sampled latent-time representations. Across regular, irregular, partially observed, and spatio-temporal benchmarks, SHIFT improves over KRNO in several settings, although the gains remain dataset- and variable-dependent.

1 INTRODUCTION

Classical statistical and deep forecasting models remain effective but typically rely on fixed or deterministic temporal representations (Milani et al., 2017; Roberts et al., 2013; Hochreiter & Schmidhuber, 1997). Recent forecasting methods address irregular and continuous-time data through adaptive representations, continuous dynamics, and probabilistic prediction (Zhang et al., 2024; Chen et al., 2023; Yalavarthi et al., 2025). In these approaches, time is typically represented through observed timestamps, continuous-time inputs, or adaptive temporal features. SHIFT instead conditions a distribution over monotone temporal reparameterizations on the observed history and samples from this distribution inside the forecasting operator.

Neural operators provide a function-space framework for temporal and spatio-temporal modeling (Lu et al., 2021; Li et al., 2021b; Kovachki et al., 2023). However, conventional temporal operators typically retain a fixed temporal coordinate system, leaving the allocation of temporal resolution deterministic and uncertainty associated with alternative temporal representations unmodeled.

SHIFT addresses this limitation through an input-conditioned stochastic monotone transformation from physical time to a normalized latent-time domain. The transformation preserves chronological ordering while allowing informative temporal regions to be expanded and others to be compressed. Within neural-operator forecasting, the closest baseline is the Khatri–Rao Neural Operator (KRNO), which formulates forecasting through a continuous time-shift operator parameterized by non-stationary integral transforms (Dama et al., 2025). SHIFT retains this operator-theoretic history-to-future forecasting perspective but augments it with an input-conditioned stochastic monotone reparameterization of time. For each observed history, SHIFT samples chronology-preserving latent-time mappings that modify the temporal representation processed by the forecasting opera-

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

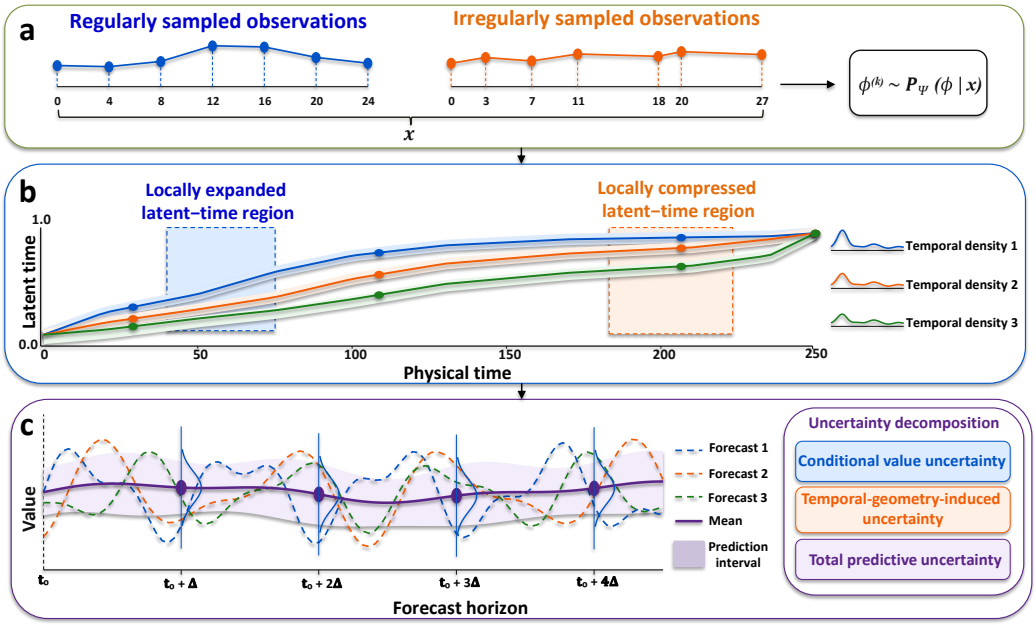


Figure 1: Conceptual overview of SHIFT. **(a)** Regularly or irregularly sampled observations condition a distribution over stochastic monotone temporal reparameterizations. **(b)** Sampled maps transform physical time to normalized latent time, locally expanding or compressing temporal regions; their derivatives define the corresponding temporal densities. **(c)** Forecasts across sampled temporal geometries are aggregated to obtain the predictive mean and prediction interval, with total uncertainty decomposed into conditional value and temporal-geometry-induced components.

tor. The resulting conditional forecasts are then aggregated across sampled temporal geometries, allowing the model to quantify forecast sensitivity to alternative learned latent-time representations.

Relation to learned temporal warping. Prior work learns monotone warps for functional registration (e.g., DeepFRC (Jiang et al., 2026)) and models uncertainty over temporal alignment (e.g., Bayesian synchronization (Aquino-López et al., 2026)). Unlike alignment methods, SHIFT uses input-conditioned stochastic mappings as internal neural-operator forecasting representations, rather than seeking a reference alignment or a uniquely identifiable physical clock. Its contribution is to propagate variation across sampled temporal maps to the predictive distribution.

SHIFT resamples each encoded history once onto a uniform latent-time grid, processes it using adaptively gated multi-scale spectral–temporal branches, and aggregates forecasts across sampled maps into conditional-value and temporal-geometry-induced uncertainty components (Figure 1).

Across regular, irregular, partially observed, and spatio-temporal forecasting benchmarks, SHIFT improves over KRNO in several settings while exhibiting dataset- and variable-dependent gains. These results indicate that stochastic temporal geometry can provide a useful inductive bias beyond explicitly irregular observations, while its benefit depends on the structure of the forecasting task. The main contributions of this work are:

- We develop an input-conditioned stochastic monotone temporal reparameterization mechanism for neural-operator forecasting. Each observed history conditions a distribution over chronology-preserving mappings from physical time to latent time, and sampled mappings modify the latent representation processed by the forecasting operator.
- We realize the reparameterization through normalized positive interval rates, yielding a differentiable order-preserving construction that supports data-dependent temporal expansion and compression and pathwise training.
- We propagate sampled temporal reparameterizations through the forecasting model and quantify forecast sensitivity to these alternative latent-time representations using the

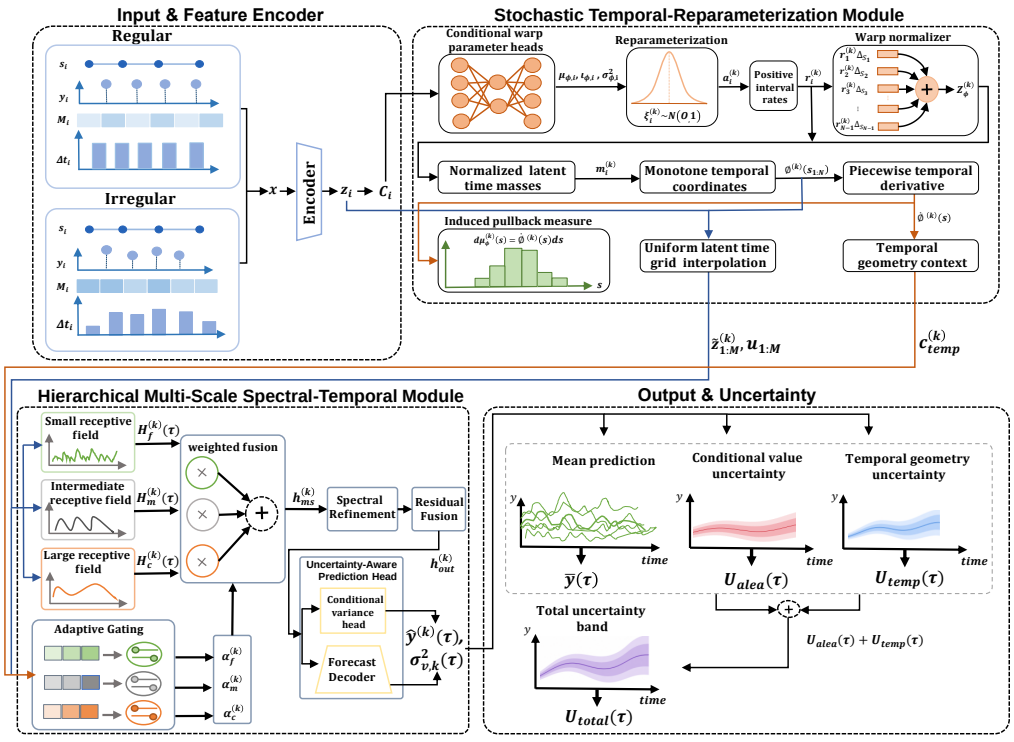


Figure 2: Overview and data flow of SHIFT. Regular or irregular observations are encoded into latent features and interval-level contexts that condition a stochastic monotone temporal reparameterization. Sampled positive interval rates are normalized to define the monotone map $\phi^{(k)}$, which resamples the encoded sequence once onto a uniform latent-time grid and induces $d\mu_\phi(s) = \dot{\phi}(s)\, ds$. The resampled representation is processed by adaptive multi-scale spectral–temporal branches and decoded into a conditional forecast mean and variance. Multiple sampled geometries are then aggregated to obtain the predictive mean and the conditional value, temporal-geometry-induced, and total predictive uncertainties.

between-reparameterization variance of conditional forecast means. Together with the conditional predictive variance, this yields the model's law-of-total-variance decomposition.

- We integrate stochastic temporal reparameterization with adaptive multi-scale spectral–temporal processing and evaluate the resulting model across regular, irregular, partially observed, and spatio-temporal forecasting tasks. Fixed-coordinate and deterministic-warp controls distinguish gains due to the broader SHIFT forecasting architecture from those attributable to learned and stochastic temporal reparameterization, revealing that the relative contribution of temporal geometry is dataset-dependent.

A broader discussion of related forecasting, temporal-warping, registration, and uncertainty-quantification methods is provided in Appendix A.

2 METHOD

SHIFT uses an input-conditioned stochastic monotone map $\phi_x : [t_p, t] \to [0, 1]$, which induces the pullback measure $d\mu_{\phi_x}(s) = \dot{\phi}_x(s)\, ds$. Encoded observations are resampled once onto the resulting normalized latent-time grid and processed by adaptive multi-scale spectral–temporal branches. Figure 2 summarizes the complete architecture. Further implementation details are provided in Appendix C.

2.1 Problem Formulation

We consider a processed observation sequence $x = \{(s_i, y_i, M_i, \Delta t_i)\}_{i=1}^{N}$, where s_i denotes the observation time, y_i the observed value, M_i the observation mask, and Δt_i the corresponding time-gap feature. SHIFT maps the observed history over $[t_p, t]$ to future values over $(t, t_f]$. The complete formulation, including endpoint assumptions, processed-time ordering, missing-value handling, and the spatio-temporal extension, is provided in Appendix C.1.

2.2 Stochastic Temporal Reparameterization and Induced Measure

Let $v : [t_p, t] \to \mathbb{R}^{d_v}$ denote the encoded history. A conventional temporal operator layer may be written as

$$(Kv)(\tau) = \int_{t_p}^{t} \kappa(\tau, s) v(s)\, ds. \tag{1}$$

SHIFT samples an input-conditioned monotone map

$$\phi_x : [t_p, t] \to [0, 1], \qquad \dot{\phi}_x(s) > 0 \quad \text{for almost every } s \in [t_p, t]. \tag{2}$$

The map is strictly increasing and absolutely continuous. In the discrete implementation, it is represented by a piecewise-linear function and is therefore differentiable everywhere except possibly at the observation times s_i. The corresponding continuous-time formulation and the full change-of-variables derivation are provided in Appendix B.

Discrete implementation. For each sampled temporal map, the encoded observations are interpolated once onto a uniform latent-time grid. The operator then acts directly on this resampled representation, so no additional Jacobian factor is applied in the latent-grid quadrature. Complete interpolation and quadrature equations are provided in Appendix C.3.

2.3 Monotone Stochastic Warp Construction

For consecutive observation times, define $\Delta s_i = s_{i+1} - s_i > 0$. Given the interval conditioning context C_i, SHIFT parameterizes a stochastic latent warp variable and constructs a positive interval rate as

$$a_i = \mu_{\phi,i} + \sigma_{\phi,i}\xi_i, \qquad \xi_i \sim \mathcal{N}(0, 1), \qquad r_i = \mathrm{softplus}(a_i). \tag{3}$$

This reparameterized sampling provides pathwise gradients through the stochastic temporal warp. The sampling introduces Monte Carlo variability in the gradient estimates; we do not employ additional control-variate estimators. The positive rates are converted into normalized latent-time masses, which determine the cumulative temporal coordinates:

$$m_i = \frac{r_i \Delta s_i}{\sum_{j=1}^{N-1} r_j \Delta s_j}, \qquad \phi(s_1) = 0, \qquad \phi(s_{i+1}) = \phi(s_i) + m_i. \tag{4}$$

Because $r_i > 0$ and $\Delta s_i > 0$, Equation 4 implies $m_i > 0$ and $\sum_i m_i = 1$. Hence, the resulting temporal map is strictly monotone and satisfies $\phi(s_1) = 0$ and $\phi(s_N) = 1$. Further details on the warp parameterization and implementation are provided in Appendix C.2.

2.4 Hierarchical Multi-Scale Spectral–Temporal Module

Three parallel temporal branches process the warped latent sequence at fine, intermediate, and coarse scales, using kernel sizes $\{3, 5, 9\}$. Input-dependent softmax gating combines their outputs using both the warped latent representation and the sampled temporal geometry. Frequency-domain refinement and learnable residual fusion are then applied. Full multiscale architecture details are provided in Appendix C.4.

2.5 Prediction Head and Conditional Value Uncertainty

For each sampled temporal geometry $\phi^{(k)}$, the decoder produces a conditional predictive mean $\hat{y}^{(k)}$ and a positive conditional variance $\sigma^2_{v,k}$, defining

$$p_\omega\Big(y \mid x, \phi^{(k)}\Big) = \mathcal{N}\Big(y; \hat{y}^{(k)}, \sigma^2_{v,k}\Big). \tag{5}$$

The Monte Carlo average of the conditional variances forms the value-uncertainty component used below. Complete decoder, variance-head, and spatio-temporal prediction equations are provided in Appendix C.7.

2.6 Temporal-Geometry-Induced Uncertainty

SHIFT treats temporal geometry as a random variable conditioned on the observed history and draws K sampled warpings $\phi^{(k)} \sim P_\psi(\phi \mid x)$. Each sampled geometry produces a conditional forecast mean $\hat{y}^{(k)}_d$ and conditional predictive variance $\sigma^2_{v,k,d}$. For a forecast component d, the predictive mean and the two modeled uncertainty components are estimated as

$$\bar{y}_d = \frac{1}{K}\sum_{k=1}^{K} \hat{y}^{(k)}_d, \qquad U_{\mathrm{alea},d} = \frac{1}{K}\sum_{k=1}^{K} \sigma^2_{v,k,d}, \qquad U_{\mathrm{temp},d} = \frac{1}{K}\sum_{k=1}^{K} \Big(\hat{y}^{(k)}_d - \bar{y}_d\Big)^2. \tag{6}$$

The total predictive uncertainty represented by SHIFT is

$$U_{\mathrm{total},d} = U_{\mathrm{alea},d} + U_{\mathrm{temp},d}. \tag{7}$$

Here, U_{alea} is the Monte Carlo average of the conditional predictive variances produced for fixed sampled temporal geometries. We retain the subscript "alea" as shorthand for this conditional value-uncertainty component; it should not be interpreted as isolating irreducible observation noise, since the conditional variance head may also absorb model mismatch and other unrepresented sources of predictive uncertainty. The term U_{temp} is the between-reparameterization variance of the conditional forecast means. Operationally, it measures how strongly the forecast changes under alternative temporal reparameterizations sampled by the learned conditional model. It therefore quantifies predictive sensitivity to stochastic latent-time geometry, rather than posterior uncertainty about a uniquely identifiable physical clock. The decomposition follows directly from the law of total variance for the represented finite predictive mixture. Further details are provided in Appendix C.8.

2.7 Training Objective

The benchmark data and evaluation protocols are inherited from the corresponding KRNO experiments, whereas the optimization objective is allowed to reflect the dataset-specific forecasting criterion together with the additional components introduced by SHIFT. For dataset D, we write

$$\mathcal{L}^{(D)}_{\mathrm{total}} = \mathcal{L}^{(D)}_{\mathrm{forecast}} + w^{(D)}_{\mathrm{reg}}\mathcal{R}^{(D)}_{\mathrm{warp}} + \mathcal{L}^{(D)}_{\mathrm{aux}}. \tag{8}$$

where $\mathcal{L}^{(D)}_{\mathrm{forecast}}$ denotes the dataset-specific forecasting objective, $\mathcal{R}^{(D)}_{\mathrm{warp}}$ contains the active temporal-geometry regularization terms, and $\mathcal{L}^{(D)}_{\mathrm{aux}}$ collects any additional dataset-specific training terms not already included in $\mathcal{L}^{(D)}_{\mathrm{forecast}}$. When conditional Gaussian NLL is active, it trains the conditional predictive variance for each sampled temporal geometry rather than the likelihood of the finite Monte Carlo predictive mixture. We study the latter objective in Appendix F.3. Exact dataset-specific objectives and regularization coefficients, including those for Crypto, are reported in Appendix E. All parameters of the encoder, stochastic temporal-reparameterization module, multi-scale spectral–temporal module, adaptive gating module, and prediction head are optimized jointly.

2.8 Inference and Interpretation

At inference time, SHIFT samples multiple temporal geometries from the learned conditional warp distribution and aggregates the resulting conditional forecasts to obtain the predictive mean and uncertainty components described in Section 2.6. The learned temporal geometry is interpreted as

Table 1: Test forecasting errors with MSE/MAE scaled by 10^{-2} for MIMIC-III, 10^{-1} for USHCN, and $10^{-3}/10^{-2}$ for Human Activity. SHIFT and KRNO results are obtained from our direct evaluations under the corresponding benchmark configurations; all other baseline values are taken from the cited benchmark studies.

		MIMIC-III		USHCN		HUMAN ACTIVITY	
METHOD	**SOURCE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**
DLinear	AAAI 2023	4.90 ± 0.00	16.29 ± 0.05	6.21 ± 0.00	3.88 ± 0.02	4.03 ± 0.01	4.21 ± 0.01
TimesNet	ICLR 2023	5.88 ± 0.08	13.62 ± 0.07	5.58 ± 0.05	3.60 ± 0.04	3.12 ± 0.01	3.56 ± 0.02
PatchTST	ICLR 2023	3.78 ± 0.03	12.43 ± 0.10	5.75 ± 0.01	3.57 ± 0.02	4.29 ± 0.14	4.80 ± 0.09
Crossformer	ICLR 2023	2.65 ± 0.10	9.56 ± 0.29	*5.25 ± 0.04*	3.27 ± 0.09	4.29 ± 0.20	4.89 ± 0.17
MTGNN	KDD 2020	2.71 ± 0.23	9.55 ± 0.65	5.39 ± 0.05	3.34 ± 0.02	3.03 ± 0.03	3.53 ± 0.03
StemGNN	NeurIPS 2020	*1.73 ± 0.02*	7.71 ± 0.11	5.75 ± 0.09	3.40 ± 0.09	8.81 ± 0.37	6.90 ± 0.02
CrossGNN	NeurIPS 2023	2.95 ± 0.16	10.82 ± 0.21	5.66 ± 0.04	3.53 ± 0.05	3.03 ± 0.10	3.48 ± 0.08
FourierGNN	NeurIPS 2023	2.55 ± 0.03	10.22 ± 0.08	5.82 ± 0.06	3.62 ± 0.07	2.99 ± 0.02	3.42 ± 0.02
SeFT	ICML 2020	1.87 ± 0.01	7.84 ± 0.08	5.80 ± 0.19	3.70 ± 0.11	12.20 ± 0.17	8.43 ± 0.05
RainDrop	ICLR 2022	1.99 ± 0.03	8.27 ± 0.07	5.78 ± 0.22	3.67 ± 0.17	14.92 ± 0.14	9.45 ± 0.05
Warpformer	KDD 2023	*1.73 ± 0.04*	*7.58 ± 0.13*	*5.25 ± 0.05*	3.23 ± 0.05	*2.79 ± 0.04*	*3.39 ± 0.03*
mTAND	ICLR 2021	1.85 ± 0.06	7.73 ± 0.13	5.33 ± 0.05	3.26 ± 0.10	3.22 ± 0.07	3.81 ± 0.07
Latent ODE	NeurIPS 2019	1.89 ± 0.19	8.11 ± 0.52	5.62 ± 0.03	3.60 ± 0.12	3.34 ± 0.11	3.94 ± 0.12
CRU	ICML 2022	1.97 ± 0.02	7.93 ± 0.19	6.09 ± 0.17	3.54 ± 0.18	6.97 ± 0.78	6.30 ± 0.47
Neural Flow	NeurIPS 2021	1.87 ± 0.05	8.03 ± 0.19	5.35 ± 0.05	3.25 ± 0.05	4.05 ± 0.13	4.36 ± 0.09
T-PatchGNN	ICML 2024	$\underline{1.69 \pm 0.03}$	$\mathbf{7.22 \pm 0.09}$	$\underline{5.00 \pm 0.04}$	*3.08 ± 0.04*	$\mathbf{2.66 \pm 0.03}$	$\mathbf{3.15 \pm 0.02}$
KRNO	ICML 2025	$\mathbf{1.57 \pm 0.02}$	$\underline{7.43 \pm 0.06}$	$\mathbf{4.95 \pm 0.08}$	$\underline{3.06 \pm 0.08}$	2.85 ± 0.03	3.46 ± 0.02
SHIFT (Ours)	-	1.979 ± 0.03	7.667 ± 0.09	$\mathbf{4.95 \pm 0.03}$	$\mathbf{3.00 \pm 0.01}$	$\underline{2.77 \pm 0.03}$	$\underline{3.29 \pm 0.01}$

a predictive latent representation rather than as recovery of a uniquely identifiable physical clock. Complete inference equations and the corresponding identifiability discussion are provided in Appendix C.

3 Experiments

We evaluate SHIFT on MIMIC-III, USHCN, Human Activity, MuJoCo, M4, Crypto, Player Trajectory, and Shallow Water, covering regular, irregular, partially observed, and spatio-temporal forecasting. The experiments assess forecasting accuracy, stochastic versus deterministic temporal reparameterization, and the contributions of the multi-scale, gating, warping, and regularization components. KRNO is rerun directly, whereas other baseline values follow the corresponding benchmark studies and are interpreted as protocol-matched comparisons. Complete dataset descriptions, baselines, metrics, preprocessing, implementation settings, and computational-cost measurements are provided in Appendix E.

3.1 Main Results

SHIFT and KRNO are evaluated using our benchmark configurations, while the remaining baseline values are taken from the cited studies and are interpreted as protocol-matched comparisons. Results are reported as mean $\pm$ standard deviation when available; bold, underlined, and italic entries denote the first-, second-, and third-best means among the methods shown, respectively. Complete reporting and comparison details are provided in Appendix E.3.

Irregular multivariate forecasting. Table 1 shows mixed performance: SHIFT matches the best reported USHCN MSE, achieves the lowest reported USHCN MAE, improves over KRNO on Human Activity, and does not improve over KRNO on MIMIC-III. Detailed comparisons are provided in Appendix F.1.1.

Robustness to missing observations. Table 2 evaluates SHIFT on MuJoCo under regular sampling and under 30%, 50%, and 70% randomly dropped observations. The continuous-time and Neural SDE baseline values are reported by Oh et al. (2024), while the KRNO values are obtained from our reruns under the corresponding benchmark protocol. SHIFT achieves the lowest MSE in

Table 2: MuJoCo test MSE with regular and randomly dropped observations.

METHOD	SOURCE	REGULAR	30% DROPPED	50% DROPPED	70% DROPPED
GRU-Δt	ICLR 2024	0.223 ± 0.020	0.198 ± 0.036	0.193 ± 0.015	0.196 ± 0.028
ACE-NODE	NeurIPS 2021	0.039 ± 0.003	0.053 ± 0.007	0.053 ± 0.005	0.052 ± 0.006
NCDE	NeurIPS 2020	0.028 ± 0.002	0.027 ± 0.000	0.027 ± 0.001	0.026 ± 0.001
ANCDE	KIS 2024	0.026 ± 0.001	0.025 ± 0.001	0.025 ± 0.001	0.024 ± 0.001
EXIT	WWW 2022	0.026 ± 0.000	0.025 ± 0.004	0.026 ± 0.000	0.026 ± 0.001
LEAP	AAAI 2023	0.022 ± 0.002	0.022 ± 0.001	0.022 ± 0.002	0.022 ± 0.001
Neural SDE	AISTATS 2020	0.028 ± 0.004	0.029 ± 0.001	0.029 ± 0.001	0.027 ± 0.000
Neural LSDE	ICLR 2024	0.013 ± 0.000	0.014 ± 0.001	0.014 ± 0.000	*0.013 ± 0.001*
Neural LNSDE	ICLR 2024	*0.012 ± 0.001*	0.014 ± 0.001	0.014 ± 0.001	0.014 ± 0.000
Neural GSDE	ICLR 2024	0.013 ± 0.001	*0.013 ± 0.001*	*0.013 ± 0.000*	0.014 ± 0.000
KRNO	ICML 2025	$\underline{0.007 \pm 0.002}$	$\underline{0.008 \pm 0.002}$	$\underline{0.0114 \pm 0.004}$	$\underline{0.0115 \pm 0.002}$
SHIFT (Ours)	-	$\mathbf{0.0018 \pm 0.0005}$	$\mathbf{0.0020 \pm 0.0007}$	$\mathbf{0.0019 \pm 0.0006}$	$\mathbf{0.0019 \pm 0.0005}$

Table 3: Crypto and Player Trajectory forecasting results. Crypto WRMSE is scaled by 10^{-3}.

METHOD	SOURCE	CRYPTO WRMSE	PLAYER RMSE
VARIMA	JEP 2001	8.76 ± 0.00	1.26 ± 0.00
MLP+RevIN+TB	ICLR 2023	7.01 ± 0.08	1.48 ± 0.25
RF+TB	ICLR 2023	7.84 ± 0.04	2.40 ± 0.01
FEDformer	ICML 2022	7.46 ± 0.04	1.29 ± 0.03
LEM	ICLR 2022	7.02 ± 0.04	1.42 ± 0.02
VBS	NeurIPS 2021	19.52 ± 0.00	5.60 ± 0.00
KNF	ICLR 2023	$\underline{6.91 \pm 0.01}$	$\underline{1.16 \pm 0.01}$
KRNO	ICML 2025	*6.95 ± 0.16*	*1.25 ± 0.05*
SHIFT (Ours)	-	$\mathbf{2.98 \pm 0.13}$	$\mathbf{0.29 \pm 0.10}$

all evaluated MuJoCo observation regimes and remains comparatively stable as the fraction of missing observations increases. Detailed numerical comparisons and a visual analysis of robustness to missing observations are provided in Appendix F.1.2.

Regular multivariate forecasting. Table 3 presents the results on Crypto and Player Trajectory. The KRNO values are obtained from our reruns under the corresponding benchmark protocol, while the remaining previously reported baseline values follow the cited benchmark studies. SHIFT achieves the lowest reported errors on both Crypto and Player Trajectory, indicating that the benefit of the complete architecture is not restricted to irregularly sampled observations. Detailed numerical comparisons and relative improvements are provided in Appendix F.1.3.

M4 forecasting. Table 4 shows that SHIFT obtains lower sMAPE than the KRNO rerun on all three evaluated M4 subsets and achieves the lowest reported sMAPE on Weekly. Detailed numerical comparisons and the reported uncertainty values, including the $K_{\text{eval}} = 1$ caveat for Quarterly, are provided in Appendix F.1.4.

Spatio-temporal forecasting. Table 5 shows a variable-dependent effect on the shallow-water benchmark: SHIFT achieves the lowest reported error for the fluid-height field ρ, while KRNO remains stronger on the two velocity components u and v. Detailed numerical comparisons and the identity-map test on the trained Shallow Water model are reported in Appendix F.1.5.

Uncertainty decomposition. Table 6 summarizes the uncertainty components estimated by SHIFT. These quantities are not ranked as forecasting metrics because a smaller predictive variance does not necessarily indicate better uncertainty calibration. Across the evaluated settings, the conditional value-uncertainty component generally accounts for most of the total predictive uncertainty, while the temporal-geometry-induced component is typically smaller but remains nonzero in several benchmarks. The absolute magnitude of U_{temp} should be interpreted as a property of

Table 4: M4 forecasting performance on the Quarterly, Weekly, and Daily subsets.

METHOD	SOURCE	QUARTERLY	WEEKLY	DAILY
FFORMA	IJF 2020	*9.733*	7.625	3.097
Smyl	IJF 2020	9.679	7.817	3.170
N-BEATS-I+G	ICLR 2020	**9.212**	–	–
KNF	ICLR 2023	10.008	*7.254*	**2.990**
KRNO	ICML 2025	10.503	6.934	*3.086*
SHIFT (Ours)	-	10.408	**6.252**	3.041

Table 5: Relative L_2 forecasting errors on the shallow-water benchmark.

METHOD	SOURCE	ϵ_ρ	ϵ_u	ϵ_v
FNO-3D	ICLR 2021	2.110×10^{-3}	2.6060×10^{-2}	2.6370×10^{-2}
LOCA	JMLR 2022	3.140×10^{-3}	1.52210×10^{-1}	1.49990×10^{-1}
KRNO	ICML 2025	1.450×10^{-3}	$\mathbf{1.4970 \times 10^{-2}}$	$\mathbf{1.4590 \times 10^{-2}}$
SHIFT (Ours)	-	$\mathbf{4.18 \times 10^{-4}}$	2.3332×10^{-2}	2.2177×10^{-2}

the trained predictive model rather than an objective-independent property of the dataset, since the training likelihood can affect agreement among sampled conditional forecasts. Overall, SHIFT's gains are dataset- and variable-dependent. Further discussion of uncertainty results and objective sensitivity is provided in Sections F.2 and F.3, respectively, while failure cases are discussed in Appendix G.

3.2 Ablation Study

We evaluate five reduced SHIFT variants on USHCN, MuJoCo with 30% dropped observations, Crypto, and Player Trajectory to isolate the effects of multi-scale processing, adaptive gating, stochastic temporal warping, learned temporal reparameterization, and smoothness regularization. Full SHIFT achieves the lowest mean forecasting error across all reported metrics, while removing any component degrades performance. Component importance is dataset-dependent: multi-scale processing has the largest effect on USHCN and Crypto, learned temporal reparameterization on MuJoCo, and smoothness regularization on Player Trajectory. The fixed-coordinate and deterministic-warp controls further show that gains over KRNO arise jointly from the broader SHIFT architecture and learned temporal geometry: temporal reparameterization provides substantial additional improvements on MuJoCo and Player Trajectory but smaller gains on USHCN and Crypto. Complete variant definitions, uncertainty ablations, and component-wise analyses are provided in Appendix F.5.

3.3 Probabilistic Calibration

We further evaluate SHIFT's probabilistic forecasts using the finite-mixture negative log-likelihood (Mix-NLL), Gaussian-mixture continuous ranked probability score (Mix-CRPS), the corresponding 95% interval score and empirical coverage computed from predictive-mixture quantiles, and mean absolute calibration error. These metrics evaluate the finite Monte Carlo predictive mixture induced by the sampled temporal geometries directly. Metric definitions and implementation details are provided in Appendix E.4. Table 8 reports results for Crypto and USHCN. Because matched probabilistic baseline reruns are unavailable, these results are descriptive. Additional analyses and limitations are discussed in Appendices F.4, G, and H.

Table 6: Uncertainty decomposition across datasets and settings. MuJoCo percentages indicate dropped observations.

SETTING	TEMPORAL UNCERTAINTY	ALEATORIC UNCERTAINTY	TOTAL UNCERTAINTY
Human Activity	$(1.80 \pm 2.10) \times 10^{-7}$	$(2.915 \pm 0.062) \times 10^{-3}$	$(2.915 \pm 0.062) \times 10^{-3}$
USHCN	$(1.50 \pm 0.50) \times 10^{-7}$	$(2.2355 \pm 0.3851) \times 10^{-2}$	$(2.2355 \pm 0.3851) \times 10^{-2}$
MIMIC-III	$(7.59 \pm 8.52) \times 10^{-13}$	$(4.167 \pm 0.04333) \times 10^{-3}$	$(4.167 \pm 0.04333) \times 10^{-3}$
MuJoCo Regular	$(2 \pm 1) \times 10^{-6}$	$(1.331 \pm 0.044) \times 10^{-3}$	$(1.333 \pm 0.044) \times 10^{-3}$
MuJoCo 30%	$(5 \pm 2) \times 10^{-6}$	$(1.363 \pm 0.044) \times 10^{-3}$	$(1.368 \pm 0.046) \times 10^{-3}$
MuJoCo 50%	$(7 \pm 5) \times 10^{-6}$	$(1.420 \pm 0.097) \times 10^{-3}$	$(1.427 \pm 0.102) \times 10^{-3}$
MuJoCo 70%	$(5 \pm 5) \times 10^{-6}$	$(1.386 \pm 0.021) \times 10^{-3}$	$(1.390 \pm 0.025) \times 10^{-3}$
Crypto	$(2.4 \pm 0.8) \times 10^{-5}$	$(4.839 \pm 0.702) \times 10^{-3}$	$(4.862 \pm 0.761) \times 10^{-3}$
Player Trajectory	$(2 \pm 1) \times 10^{-6}$	$(3.094 \pm 0.005) \times 10^{-3}$	$(3.097 \pm 0.005) \times 10^{-3}$
Shallow Water	1.6920×10^{-4}	1.292529×10^{-3}	1.461734×10^{-3}
M4 Quarterly	0	7.690×10^{-3}	7.690×10^{-3}
M4 Weekly	1.4×10^{-5}	5.658×10^{-3}	5.672×10^{-3}
M4 Daily	1.5×10^{-5}	4.920×10^{-3}	4.936×10^{-3}

Table 7: Attribution of forecasting gains using fixed-coordinate and deterministic-warp controls. Lower is better.

METRIC	KRNO	FIXED-COORDINATE	DETERMINISTIC WARP	SHIFT (FULL)
USHCN MSE	4.95	5.06	5.08	**4.95**
USHCN MAE	3.06	3.13	3.14	**3.00**
MuJoCo, 30% dropped MSE	0.008	0.00345	0.00329	**0.0020**
Crypto WRMSE	6.95	3.00	3.03	**2.98**
Player Trajectory RMSE	1.25	0.40	0.39	**0.29**

Table 8: Probabilistic forecasting and uncertainty calibration on Crypto and USHCN.

DATASET	METHOD	MIX-NLL	MIX-CRPS	INTERVAL SCORE	COVERAGE $_{95}$	CALIB. ERROR
Crypto	**SHIFT**	-5.2481	0.000625	0.007325	0.9011	0.0227
USHCN	**SHIFT**	1.0872	0.241554	3.602832	0.9564	0.0924

4 Conclusion

We presented SHIFT, a neural-operator forecasting framework combining input-conditioned stochastic temporal reparameterization with adaptive multi-scale spectral–temporal processing. Sampled latent-time representations provide a decomposition of predictive uncertainty into conditional value and temporal-geometry-induced components. Across regular, irregular, partially observed, and spatio-temporal benchmarks, SHIFT improves over KRNO in several settings. Controlled fixed-coordinate and deterministic-warp comparisons show that these gains arise jointly from the broader forecasting architecture and learned temporal geometry, with the contribution of stochastic temporal reparameterization remaining dataset-dependent.

Reproducibility Statement

Detailed model formulations, implementation details, dataset protocols, evaluation procedures, hyperparameters, and additional experimental results are provided in Appendices C, E, and F.5.

AI USE STATEMENT

Generative AI tools were used to assist with manuscript editing, shortening and rephrasing, and LaTeX organization. All AI-assisted content was reviewed and verified by the authors. The authors take full responsibility for the final content of the manuscript.

REFERENCES

Marco A. Aquino-López, Francesco Muschitiello, and Matt Osman. Bayesian synchronization of proxy paleorecords with reference chronologies. *arXiv preprint arXiv:2603.10218*, 2026.

Marin Biloš, Johanna Sommer, Syama Sundar Rangapuram, Tim Januschowski, and Stephan Günnemann. Neural flows: Efficient alternative to neural odes. In *Advances in Neural Information Processing Systems*, volume 34, pp. 21325–21337, 2021.

Christopher Bülte, Philipp Scholl, and Gitta Kutyniok. Probabilistic neural operators for functional uncertainty quantification. *arXiv preprint arXiv:2502.12902*, 2025.

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

Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific Reports*, 8(1):6085, 2018. doi: 10.1038/s41598-018-24271-9.

Yuqi Chen, Kan Ren, Yansen Wang, Yuchen Fang, Weiwei Sun, and Dongsheng Li. ContiFormer: Continuous-time transformer for irregular time series modeling. In *Advances in Neural Information Processing Systems*, volume 36, pp. 47143–47175, 2023.

Srinath Dama, Kevin Course, and Prasanth B. Nair. Shifting time: Time-series forecasting with Khatri–Rao neural operators. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025.

Jeffrey L. Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211, 1990.

Agathe Girard. *Approximate Methods for Propagation of Uncertainty with Gaussian Process Models*. PhD thesis, University of Glasgow, 2004.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8): 1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735.

Max Horn, Michael Moor, Christian Bock, Bastian Rieck, and Karsten Borgwardt. Set functions for time series. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 4353–4363. PMLR, 2020.

Qihe Huang, Lei Shen, Ruixin Zhang, Shouhong Ding, Binwu Wang, Zhengyang Zhou, and Yang Wang. CrossGNN: Confronting noisy multivariate time series via cross interaction refinement. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Sheo Yon Jhin, Minju Jo, Taeyong Kong, Jinsung Jeon, and Noseong Park. ACE-NODE: Attentive co-evolving neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Sheo Yon Jhin, Jaehoon Lee, Minju Jo, Seungji Kook, Jinsung Jeon, Jihyeon Hyeong, Jayoung Kim, and Noseong Park. EXIT: Extrapolation and interpolation-based neural controlled differential equations for time-series classification and forecasting. In *Proceedings of the ACM Web Conference 2022*, pp. 3102–3112, 2022.

Sheo Yon Jhin, Minju Jo, Seungji Kook, and Noseong Park. Learnable path in neural controlled differential equations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 8014–8022, 2023.

Sheo Yon Jhin, Heejoo Shin, Sujie Kim, Seoyoung Hong, Minju Jo, Solhee Park, Noseong Park, Seungbeom Lee, Hwiyoung Maeng, and Seungmin Jeon. Attentive neural controlled differential equations for time-series classification and forecasting. *Knowledge and Information Systems*, 66 (3):1885–1915, 2024.

Siyuan Jiang, Yihan Hu, Wenjie Li, and Pengcheng Zeng. Deepfrc: An end-to-end deep learning model for functional registration and classification. In *International Conference on Learning Representations*, 2026.

Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. MIMIC-III, a freely accessible critical care database. *Scientific Data*, 3(1):1–9, 2016.

Patrick Kidger, James Morrill, James Foster, and Terry Lyons. Neural controlled differential equations for irregular time series. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6696–6707, 2020.

Byunghyun Kim and Jae-Gil Lee. Continuous-time linear positional embedding for irregular time series forecasting. *arXiv preprint arXiv:2409.20092*, 2024.

Georgios Kissas, Jacob H Seidman, Leonardo Ferreira Guilhoto, Victor M Preciado, George J Pappas, and Paris Perdikaris. Learning operators with coupled attention. *Journal of Machine Learning Research*, 23(215):1–63, 2022.

Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to PDEs. *Journal of Machine Learning Research*, 24(89):1–97, 2023.

Alexander C. Li, Shiyu Chen, Josue Nassar, Suchi Saria, and Stephan Mandt. Detecting and adapting to irregular distribution shifts in bayesian online learning. In *Advances in Neural Information Processing Systems*, volume 34, pp. 6816–6828, 2021a.

Xuechen Li, Ting-Kam Leonard Wong, Ricky T. Q. Chen, and David Duvenaud. Scalable gradients for stochastic differential equations. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 3870–3882. PMLR, 2020.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021b.

Bryan Lim, Sercan O. Arik, Nicolas Loeff, and Tomas Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4): 1748–1764, 2021. doi: 10.1016/j.ijforecast.2021.03.012.

Matthew Lowery, John Turnage, Zachary Morrow, John D. Jakeman, Akil Narayan, Shandian Zhe, and Varun Shankar. Kernel neural operators (KNOs) for scalable, memory-efficient, geometrically-flexible operator learning. *arXiv preprint arXiv:2407.00809*, 2024.

Lu Lu, Pengzhan Jin, and George Em Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3): 218–229, 2021.

Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. The M4 competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting*, 36(1): 54–74, 2020.

MJ Menne, CN Williams Jr, RS Vose, and Data Files. Long-term daily climate records from stations across the contiguous united states, 2015.

Eder Milani, Marinho Andrade, and Carlos Diniz. Generalized normal arma model applied to the areas of economy, hydrology, and public policy. *Communications in Statistics - Simulation and Computation*, 46(7):5819–5835, 2017.

Pablo Montero-Manso, George Athanasopoulos, Rob J. Hyndman, and Thiyanga S. Talagala. FFORMA: Feature-based forecast model averaging. *International Journal of Forecasting*, 36 (1):86–92, 2020. doi: 10.1016/j.ijforecast.2019.02.011.

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

YongKyung Oh, Dongyoung Lim, and Sungil Kim. Stable Neural Stochastic Differential Equations in analyzing irregular time series data. In *The Twelfth International Conference on Learning Representations*, 2024.

Boris N. Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. In *International Conference on Learning Representations*, 2020.

Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8857–8868. PMLR, 2021.

Stephen Roberts, Michael Osborne, Mark Ebden, Steven Reece, Neale Gibson, and Suzanne Aigrain. Gaussian processes for time-series modelling. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1984):20110550, 2013.

Yulia Rubanova, Ricky T. Q. Chen, and David Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

T. Konstantin Rusch, Siddhartha Mishra, N. Benjamin Erichson, and Michael W. Mahoney. Long expressive memory for sequence modeling. In *International Conference on Learning Representations*, 2022.

David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3): 1181–1191, 2020. doi: 10.1016/j.ijforecast.2019.07.001.

Mona Schirmer, Mazin Eltayeb, Stefan Lessmann, and Maja Rudolph. Modeling irregular time series with continuous recurrent units. In *International Conference on Machine Learning*, pp. 19388–19405. PMLR, 2022.

Satya Narayan Shukla and Benjamin M. Marlin. Multi-time attention networks for irregularly sampled time series. In *International Conference on Learning Representations*, 2021.

Slawek Smyl. A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. *International Journal of Forecasting*, 36(1):75–85, 2020. doi: 10.1016/j.ijforecast.2019.03.017.

James H. Stock and Mark W. Watson. Vector autoregressions. *Journal of Economic Perspectives*, 15(4):101–115, 2001. doi: 10.1257/jep.15.4.101.

Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. CSDI: Conditional score-based diffusion models for probabilistic time series imputation. In *Advances in Neural Information Processing Systems*, volume 34, pp. 24804–24816, 2021.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. DeepMind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

Alessandro Ticchi, Andrew Scherer, Carla McIntyre, Carlos Stein N Brito, Derek Snow, Develra, dstern, James Colless, Kieran Garvey, Maggie, Maria Perez Ortiz, Ryan Lynch, and Sohier Dane. G-research crypto forecasting. `https://kaggle.com/competitions/g-research-crypto-forecasting`, 2021. Kaggle.

Rui Wang, Yihe Dong, Sercan O. Arik, and Rose Yu. Koopman neural forecaster for time series with temporal distribution shifts. In *International Conference on Learning Representations*, 2023.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. TimesNet: Temporal 2d-variation modeling for general time series analysis. In *International Conference on Learning Representations*, 2023.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 753–763, 2020.

Vijaya Krishna Yalavarthi, Randolf Scholz, Stefan Born, and Lars Schmidt-Thieme. Probabilistic forecasting of irregularly sampled time series with missing values via conditional normalizing flows. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 21877–21885, 2025.

Kun Yi, Qi Zhang, Wei Fan, Hui He, Liang Hu, Pengyang Wang, Ning An, Longbing Cao, and Zhendong Niu. Fouriergnn: Rethinking multivariate time series forecasting from a pure graph perspective. *Advances in neural information processing systems*, 36:69638–69660, 2023.

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

Jiawen Zhang, Shun Zheng, Wei Cao, Jiang Bian, and Jia Li. Warpformer: A multi-scale modeling approach for irregular clinical time series. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 3273–3285, 2023.

Weijia Zhang, Chenlong Yin, Hao Liu, Xiaofang Zhou, and Hui Xiong. Irregular multivariate time series forecasting: A transformable patching graph neural networks approach. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Xiang Zhang, Marko Zeman, Theodoros Tsiligkaridis, and Marinka Zitnik. Graph-guided network for irregularly sampled multivariate time series. In *International Conference on Learning Representations*, 2022.

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

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 27268–27286. PMLR, 2022.

A Extended Related Work

A.1 Forecasting with Regular and Irregular Observations

A.1.1 Irregular Forecasting.

Irregular time-series forecasting is challenging because observations may be unevenly sampled, partially missing, asynchronous, or affected by variable time gaps. These issues are especially important in real-world domains such as healthcare, climate, finance, and physical systems, where the absence of an observation or the length of a time gap can itself carry predictive information. Early recurrent approaches such as GRU-D incorporate missingness patterns using observation masks and time intervals, showing that missing values and their observation patterns should not be treated only as preprocessing issues (Che et al., 2018). Continuous-time models such as ODE-RNN and Latent ODE define hidden dynamics through ordinary differential equations, allowing the model to handle arbitrary observation times and irregular temporal intervals (Rubanova et al., 2019).

More recent methods improve irregular time-series forecasting using adaptive temporal representations and continuous-time architectures. T-PatchGNN models irregular multivariate time series using transformable patches and time-adaptive graph neural networks to capture local temporal semantics and dynamic inter-series correlations (Zhang et al., 2024). ContiFormer combines continuous-time dynamics with Transformer-style attention for irregular time-series modeling (Chen et al., 2023). Continuous-Time Linear Positional Embedding represents irregular timestamps using learnable continuous-time functions (Kim & Lee, 2024). ProFITi performs probabilistic forecasting with missing and irregular observations using conditional normalizing flows, learning flexible multivariate conditional distributions without assuming a fixed output distribution (Yalavarthi et al., 2025). These methods demonstrate the importance of modeling non-uniform sampling, asynchronous observations, missing values, and time gaps. However, they generally still treat time as an observed coordinate, positional encoding, patching variable, or input to a continuous-time model. SHIFT instead learns an input-conditioned stochastic monotone reparameterization of the observed temporal coordinates. The induced pullback measure provides an equivalent representation in physical time, whereas the implemented architecture interpolates the encoded observations onto a normalized latent-time grid.

A.1.2 Regular Forecasting.

Regular time-series forecasting assumes that observations are collected on a fixed temporal grid. Classical methods, including autoregressive and moving-average models, remain widely used because of their interpretability and statistical grounding. Generalized normal ARMA models relax Gaussian innovation assumptions by using generalized normal distributions with flexible tail behavior (Milani et al., 2017), while Gaussian processes provide a Bayesian framework for uncertainty-aware temporal modeling and multi-step uncertainty propagation (Roberts et al., 2013; Girard, 2004). Deep learning methods further improve temporal representation: recurrent neural networks model temporal dependencies through recurrent hidden states (Elman, 1990), and Long Short-Term Memory networks capture longer dependencies using gated memory mechanisms (Hochreiter & Schmidhuber, 1997). DeepAR extends recurrent forecasting to probabilistic prediction by learning autoregressive predictive distributions across related time series (Salinas et al., 2020), while Temporal Fusion Transformers combine recurrent local processing, attention-based long-range dependency modeling, feature selection, and gating mechanisms for interpretable multi-horizon forecasting (Lim et al., 2021).

Neural operators provide a function-space framework for learning mappings between temporal or spatio-temporal functions. DeepONet learns nonlinear operators from data (Lu et al., 2021), Fourier Neural Operators use Fourier-domain integral operators for solution learning (Li et al., 2021b), and neural operator theory provides a foundation for discretization-invariant learning (Kovachki et al., 2023). Kernel Neural Operators improve geometric flexibility by decoupling trainable kernels from numerical integration schemes (Lowery et al., 2024). Within neural-operator forecasting, the closest baseline considered here is the Khatri–Rao Neural Operator (KRNO), which introduces a continuous time-shift operator using non-stationary integral transforms with a Khatri–Rao product structure (Dama et al., 2025). Relative to this forecasting baseline, SHIFT uses an input-conditioned stochastic temporal reparameterization that is shared by interpolation, multi-scale representation, and pre-

dictive sampling. This mechanism is also applicable to regularly sampled data because equal spacing in physical time does not necessarily imply equal predictive importance. Accordingly, SHIFT adapts the allocation of latent temporal resolution in both regular and irregular forecasting.

A.2 Learned Temporal Warping and Functional Registration

Temporal warping has also been studied in functional registration and alignment, where temporal coordinates are transformed so that corresponding structures become better aligned. Recent neural approaches learn monotone or diffeomorphic warpings jointly with a downstream task. For example, DeepFRC jointly learns diffeomorphic functional registration and classification using a neural warping operator and spectral representations (Jiang et al., 2026). Bayesian synchronization methods have additionally considered posterior uncertainty over monotone mappings when aligning observations recorded on different temporal scales (Aquino-López et al., 2026).

These directions are closely related to SHIFT in their use of chronology-preserving temporal transformations, but the modeling objective is different. SHIFT does not attempt to register an observed sequence to a known reference trajectory or recover a uniquely identifiable alignment. Instead, an input-conditioned stochastic monotone reparameterization is used internally within the forecasting operator. The sampled reparameterizations modify the latent representation supplied to the forecasting architecture, and their induced variation is propagated into the predictive distribution. Thus, SHIFT uses stochastic temporal reparameterization as a forecasting mechanism rather than temporal registration as an independently supervised alignment task.

A.3 Uncertainty Quantification for Time Series

Uncertainty quantification is important for time-series forecasting under noise, missing observations, irregular sampling, temporal misalignment, and long-horizon prediction. Gaussian processes provide a classical Bayesian framework for predictive uncertainty in time-series modeling (Roberts et al., 2013), and uncertainty propagation has been studied for multi-step forecasting with Gaussian process models (Girard, 2004). In deep forecasting, DeepAR learns autoregressive predictive distributions (Salinas et al., 2020), while Temporal Fusion Transformers support multi-horizon forecasting with quantile-based predictive intervals (Lim et al., 2021). Diffusion-based models such as TimeGrad generate multivariate probabilistic forecasts through autoregressive denoising diffusion processes (Rasul et al., 2021). CSDI uses conditional score-based diffusion for probabilistic time-series imputation and related forecasting settings (Tashiro et al., 2021). For irregular and missing observations, ProFITi models flexible multivariate predictive distributions using conditional normalizing flows (Yalavarthi et al., 2025). In operator learning, Probabilistic Neural Operators learn distributions over output function spaces for functional uncertainty quantification (Bülte et al., 2025).

Although uncertainty-aware forecasting has been studied in several forms, most existing approaches focus primarily on uncertainty in output values, predictive distributions, model parameters, or missing-value imputation. Comparatively less attention has been given to predictive variability arising from alternative temporal representations, particularly within neural-operator forecasting.

Within SHIFT, stochastic temporal reparameterization provides an additional modeled source of predictive variability. Different samples from the learned conditional warp distribution correspond to alternative latent-time representations under the SHIFT model. Predictive variation across these sampled reparameterizations defines the temporal-geometry-induced uncertainty component, whereas conditional output variance represents value uncertainty for a fixed sampled reparameterization. Their combination yields the law-of-total-variance decomposition used by SHIFT. This component should be interpreted as forecast sensitivity to the learned stochastic temporal representation; it does not represent posterior uncertainty over network parameters or uncertainty about a uniquely identifiable physical clock.

B THEORY AND DERIVATIONS

B.1 CONTINUOUS-TIME FORMULATION

Conceptually, let $q_x(s) > 0$ denote an input-conditioned stochastic temporal rate:

$$q_x(s) = \text{softplus}\left(g_\psi\left(z(s), M(s), \Delta t(s)\right) + \sigma_\psi(s)\xi(s)\right), \tag{9}$$

where $\xi(s)$ denotes a zero-mean, unit-variance Gaussian process. This continuous expression is used only as a conceptual analogue of the interval-wise Gaussian perturbations employed by the discrete implementation. The positivity of $q_x(s)$ guarantees that the resulting temporal map preserves chronological order.

Define the normalization constant

$$Z_\phi = \int_{t_p}^{t} q_x(u)\, du. \tag{10}$$

The normalized stochastic temporal map is then

$$\phi_x(s) = \frac{\int_{t_p}^{s} q_x(u)\, du}{\int_{t_p}^{t} q_x(u)\, du}. \tag{11}$$

Consequently,

$$\dot{\phi}_x(s) = \frac{q_x(s)}{Z_\phi} > 0 \quad \text{for almost every } s \in [t_p, t]. \tag{12}$$

Therefore,

$$\phi_x(t_p) = 0, \qquad \phi_x(t) = 1,$$

and the temporal ordering of the observations is preserved. The discrete interval-rate construction introduced below is an interval-wise approximation of this continuous probabilistic temporal geometry.

B.2 PULLBACK MEASURE AND CHANGE OF VARIABLES

For each sampled temporal map ϕ, we assume that ϕ is strictly increasing and absolutely continuous, with

$$\phi(t_p) = 0, \qquad \phi(t) = 1, \qquad \dot{\phi}(s) > 0 \quad \text{for almost every } s \in [t_p, t]. \tag{13}$$

These conditions guarantee that ϕ^{-1} exists on $[0, 1]$ and that the change-of-variables formula is valid.

For a fixed sampled temporal geometry, define the latent-time representation

$$\tilde{v}_\phi(u) = v\big(\phi^{-1}(u)\big), \qquad u \in [0, 1]. \tag{14}$$

The temporal operator is evaluated on the normalized latent-time domain as

$$(K_\phi v)(\tau) = \int_0^1 \kappa(\tau, u)\tilde{v}_\phi(u)\, du. \tag{15}$$

Applying the change of variables

$$u = \phi(s), \qquad du = \dot{\phi}(s)\, ds,$$

and using

$$\tilde{v}_\phi(\phi(s)) = v(s),$$

gives the equivalent physical-time representation

$$(K_\phi v)(\tau) = \int_{t_p}^{t} \kappa(\tau, \phi(s))\, v(s)\dot{\phi}(s)\, ds. \tag{16}$$

The sampled temporal map induces the temporal measure

$$d\mu_\phi(s) = \dot{\phi}(s)\, ds. \tag{17}$$

Therefore, the same operator may be written as

$$(K_\phi v)(\tau) = \int_{t_p}^{t} \kappa(\tau, \phi(s))\, v(s)\, d\mu_\phi(s). \tag{18}$$

Because ϕ is sampled from the input-conditioned distribution $P(\phi \mid x)$, the induced measure μ_ϕ is learned, data-dependent, and stochastic. Different samples of ϕ therefore define different plausible temporal geometries and different allocations of temporal mass over the observed history.

The latent-time integral and the physical-time integral under $d\mu_\phi$ are two mathematically equivalent representations of the same operator. The induced measure is not applied as an additional weight after latent-grid resampling; doing so would count the temporal reparameterization twice.

B.3 Monotonicity and Boundary Conditions

Since the interval rates are produced by the softplus transformation, we have $r_i > 0$. Moreover, the processed observation times are strictly increasing, so that

$$\Delta s_i = s_{i+1} - s_i > 0.$$

Because

$$Z_\phi = \sum_{j=1}^{N-1} r_j \Delta s_j,$$

we also have $Z_\phi > 0$. Therefore, for every consecutive pair of observation times,

$$\phi(s_{i+1}) - \phi(s_i) = m_i = \frac{r_i \Delta s_i}{Z_\phi} > 0. \tag{19}$$

Hence,

$$\phi(s_1) < \phi(s_2) < \cdots < \phi(s_N),$$

and the sampled temporal map strictly preserves chronological order.

The terminal boundary condition follows from the normalization:

$$\begin{aligned} \phi(s_N) &= \phi(s_1) + \sum_{i=1}^{N-1} m_i \\ &= 0 + \sum_{i=1}^{N-1} \frac{r_i \Delta s_i}{Z_\phi} \\ &= \frac{\sum_{i=1}^{N-1} r_i \Delta s_i}{Z_\phi} = 1. \end{aligned} \tag{20}$$

Thus, the sampled map sends the first observation time to 0, the last observation time to 1, and is strictly increasing over the complete observation interval.

Between consecutive observation times, the map is defined by piecewise-linear interpolation:

$$\phi(s) = \phi(s_i) + \frac{s - s_i}{\Delta s_i} \left[\phi(s_{i+1}) - \phi(s_i)\right], \qquad s \in [s_i, s_{i+1}]. \tag{21}$$

This construction gives a continuous and absolutely continuous temporal map. The map is differentiable on each open interval (s_i, s_{i+1}), while its derivative may be undefined at the interval boundaries s_i. These finitely many boundary points have zero measure and do not affect the induced measure or the change-of-variables formulation.

The corresponding derivative exists almost everywhere and is piecewise constant:

$$\dot{\phi}(s) = \frac{\phi(s_{i+1}) - \phi(s_i)}{\Delta s_i} = \frac{r_i}{Z_\phi}, \qquad s \in [s_i, s_{i+1}). \tag{22}$$

This piecewise-constant derivative is the discrete counterpart of the pullback density derived above and completely determines the induced temporal measure.

C Detailed Architecture

C.1 Input Representation and Conditioning

The mask M_i indicates which components of y_i are observed at time s_i. Numerical placeholders used for missing components are inherited from the corresponding benchmark preprocessing pipeline and are not interpreted as observations. SHIFT uses the processed values jointly with their observation masks and time-gap information. The proposed method does not modify the benchmark-specific imputation, placeholder, or mask-construction procedure.

Let an observed time series over a look-back interval $[t_p, t]$ be represented as

$$x = \{(s_i, y_i, M_i, \Delta t_i)\}_{i=1}^{N}, \tag{23}$$

where $s_i \in [t_p, t]$ is an observation time, $y_i \in \mathbb{R}^{d_y}$ is the corresponding observed value, $M_i \in \{0,1\}^{d_y}$ is an observation mask, and Δt_i is the local time-gap feature supplied by the retained benchmark pipeline. For $i \geq 2$, it is based on the elapsed time since the preceding processed observation, while its boundary value at $i = 1$ follows the corresponding benchmark convention.

The sequence supplied to SHIFT is the processed observation sequence produced by the corresponding benchmark pipeline. We index the processed observation times in strictly increasing order:

$$s_1 < s_2 < \cdots < s_N. \tag{24}$$

For the temporal-reparameterization module, the processed look-back sequence is anchored at the boundaries of the observation window. The preprocessing pipeline therefore retains temporal support points at t_p and t. If an endpoint value is unavailable because of missingness or observation dropping, its timestamp is retained and the corresponding observation mask marks the value as unobserved rather than removing the boundary support point. Accordingly,

$$s_1 = t_p, \qquad s_N = t. \tag{25}$$

This convention fixes the domain of the temporal reparameterization independently of the observation pattern and makes the endpoint conditions of the discrete and continuous temporal maps consistent. Accordingly, the consecutive physical-time intervals satisfy

$$\Delta s_i = s_{i+1} - s_i > 0, \qquad i = 1, \ldots, N-1. \tag{26}$$

This is an input requirement of the temporal-reparameterization module. Any benchmark-specific handling of repeated raw timestamps is performed by the retained preprocessing pipeline before the sequence is supplied to SHIFT. For regularly sampled data, the time gaps may be constant and the masks may indicate fully observed sequences. For irregularly sampled or partially observed data, Δt_i and M_i encode non-uniform sampling and missingness. Here, Δt_i denotes the benchmark-provided time-gap feature supplied to the encoder, whereas $\Delta s_i = s_{i+1} - s_i$ denotes the scalar physical-time interval used by the temporal-reparameterization module.

The objective is to learn an operator that maps the observed history over $[t_p, t]$ to future values over $(t, t_f]$. For a temporal forecasting problem, the operator is written as

$$\hat{y}(\tau) = \mathcal{A}_\theta(x)(\tau), \qquad \tau \in (t, t_f], \tag{27}$$

where $\mathcal{A}_\theta$ denotes the learned forecasting operator.

For a spatio-temporal field $u(a, s)$, where $a \in \Omega$ denotes a spatial coordinate, the formulation becomes

$$\hat{u}(a, \tau) = \mathcal{A}_\theta(u)(a, \tau), \qquad a \in \Omega, \quad \tau \in (t, t_f]. \tag{28}$$

In both settings, SHIFT learns an operator from a past function to a future function. The proposed distinction is not an independently parameterized temporal measure, but an input-conditioned stochastic reparameterization that controls the latent temporal coordinates used by the forecasting architecture.

C.1.1 Warp Conditioning Context

An encoder first maps each observation to a latent representation:

$$z_i = E_\theta\left(y_i, M_i, \Delta t_i, s_i\right), \qquad i = 1, \ldots, N. \tag{29}$$

We construct an interval-level conditioning context as

$$C_i = \left[z_i, z_{i+1}, M_i, M_{i+1}, \Delta s_i\right], \qquad i = 1, \ldots, N-1. \tag{30}$$

C.2 Stochastic Warp Construction

Let

$$\Delta s_i = s_{i+1} - s_i, \qquad i = 1, \ldots, N-1, \tag{31}$$

denote the lengths of the physical-time intervals between consecutive observations.

For each interval, let C_i denote a conditioning context formed from the encoded adjacent observations z_i, z_{i+1}, their masks M_i, M_{i+1}, and the interval length Δs_i. The complete encoder and interval-context construction equations are provided in Appendix C.1.1.

Two neural-network heads predict the conditional mean and log-variance of the latent interval warp variables:

$$\mu_{\phi,i} = g_{\mu,\psi}(C_i), \tag{32}$$
$$\ell_{\phi,i} = g_{\sigma,\psi}(C_i), \tag{33}$$
$$\sigma^2_{\phi,i} = \exp\left(\ell_{\phi,i}\right) + \epsilon_0, \tag{34}$$

where $\epsilon_0 > 0$ is used for numerical stability.

The learned conditional warp distribution is

$$p_\psi(a_{1:N-1} \mid x) = \prod_{i=1}^{N-1} \mathcal{N}\left(a_i; \mu_{\phi,i}, \sigma^2_{\phi,i}\right). \tag{35}$$

Here, $\sigma_{\phi,i} = \sqrt{\sigma^2_{\phi,i}}$. The corresponding standard-deviation expression is given in Eq. equation 45 below.

Samples are generated using the reparameterization trick:

$$\xi_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1), \tag{36}$$
$$a_i = \mu_{\phi,i} + \sigma_{\phi,i}\xi_i, \tag{37}$$
$$r_i = \mathrm{softplus}(a_i), \qquad i = 1, \ldots, N-1. \tag{38}$$

The positive interval rates are normalized using

$$Z_\phi = \sum_{i=1}^{N-1} r_i \Delta s_i. \tag{39}$$

In the implementation, Z_ϕ is lower-bounded by a small positive constant before division.

Define the normalized latent-time mass assigned to interval i as

$$m_i = \frac{r_i \Delta s_i}{Z_\phi}, \qquad i = 1, \ldots, N-1. \tag{40}$$

Since $r_i > 0$ and $\Delta s_i > 0$, the normalized interval masses satisfy

$$m_i > 0, \qquad \sum_{i=1}^{N-1} m_i = 1. \tag{41}$$

The shared normalization induces dependence among the normalized interval masses and makes the temporal map invariant to a common positive rescaling of all interval rates. These properties are discussed below, including the scale-invariance relation in Eq. equation 44.

The monotone temporal coordinates are constructed recursively:

$$\phi(s_1) = 0, \tag{42}$$

$$\phi(s_{i+1}) = \phi(s_i) + m_i, \qquad i = 1, \ldots, N-1. \tag{43}$$

Since $r_i > 0$ and $\Delta s_i > 0$, the resulting temporal map is strictly monotone and satisfies the boundary conditions $\phi(s_1) = 0$ and $\phi(s_N) = 1$. A complete derivation is provided in Appendix B.3.

Although the pre-normalization latent variables a_i are conditionally factorized given the input sequence, the normalized masses m_i are statistically dependent through the shared random normalizer Z_ϕ. Therefore, the sampled temporal geometry defines a jointly normalized stochastic allocation of latent time across the complete observation interval.

The absolute scale of the unnormalized positive rates is not identifiable after normalization. In particular, multiplying all rates by the same positive constant $c > 0$ gives

$$\frac{(cr_i)\Delta s_i}{\sum_{j=1}^{N-1}(cr_j)\Delta s_j} = \frac{r_i \Delta s_i}{\sum_{j=1}^{N-1} r_j \Delta s_j}. \tag{44}$$

Therefore, the normalized temporal map is determined by the relative interval rates, or equivalently by their normalized latent-time masses, rather than by the absolute magnitude of the unnormalized rates. This scale invariance applies to the normalized forecasting geometry itself. It does not imply that the complete training objective is invariant to a common rescaling of the pre-normalization rates. In particular, the rate, increment-smoothness, and optional length-consistency regularizers act on pre-normalization quantities and therefore introduce scale-dependent preferences during optimization; when active, the variational KL regularizer additionally constrains the pre-softplus interval variables. Thus, the absolute rate scale is not identifiable from the normalized temporal map, while the scale realized during training can be selected by the regularization terms.

The corresponding conditional standard deviation is

$$\sigma_{\phi,i} = \sqrt{\exp\left(\ell_{\phi,i}\right) + \epsilon_0}. \tag{45}$$

Because this distribution is conditioned only on the observed history available at inference time, we refer to it as a conditional warp distribution rather than an approximate posterior.

C.3 Latent-Time Interpolation

Discrete implementation. We use a uniform latent-time grid

$$u_j = \frac{j-1}{M-1}, \qquad j = 1, \ldots, M. \tag{46}$$

Given the sampled coordinates $\phi_i = \phi(s_i)$, the encoded observations are resampled onto the uniform latent-time grid using piecewise-linear interpolation. For each $u_j \in [\phi_i, \phi_{i+1}]$, define

$$\lambda_{ij} = \frac{u_j - \phi_i}{\phi_{i+1} - \phi_i}. \tag{47}$$

The resampled latent representation is

$$\tilde{z}_j = (1 - \lambda_{ij}) z_i + \lambda_{ij} z_{i+1}, \qquad u_j \in [\phi_i, \phi_{i+1}]. \tag{48}$$

Because the sampled temporal map satisfies $\phi(s_1) = 0$ and $\phi(s_N) = 1$, every point $u_j \in [0, 1]$ lies within the range of the sampled coordinates. Therefore, no extrapolation outside the observation interval is required. At the endpoints, the interpolation gives $\tilde{z}_1 = z_1$ and $\tilde{z}_M = z_N$.

The discrete operator is then

$$(K_\phi z)(\tau_m) \approx \sum_{j=1}^{M} \kappa(\tau_m, u_j) \tilde{z}_j \tilde{w}_j, \tag{49}$$

where $\tilde{w}_j$ are quadrature weights on the uniform latent-time grid. No additional factor of $\dot{\phi}$ is included in equation 49, because the effect of the reparameterization has already been incorporated through equation 48.

The induced measure in equation 17 provides the equivalent physical-coordinate interpretation of the same computation. It is not an additional independently learned weighting function.

C.4 Multi-Scale Temporal Processing

Temporal processes often contain interacting structures at multiple scales. Fine-scale dynamics may encode local transitions, abrupt events, or high-frequency variations, whereas coarse-scale dynamics may capture trends, long-range dependencies, periodicity, and slowly evolving latent states.

For the k-th sampled temporal geometry, the warped latent sequence is defined as

$$\tilde{z}_j^{(k)} = \mathrm{Interp}\left(\{(\phi^{(k)}(s_i), z_i)\}_{i=1}^{N}, u_j\right), \qquad j = 1, \ldots, M. \tag{50}$$

The hierarchical spectral–temporal module comprises parallel temporal branches associated with progressively larger receptive-field levels:

$$H_r^{(k)}(\tau) = \mathcal{B}_r\left(\tilde{z}^{(k)}, u_{1:M}\right)(\tau), \qquad r = 1, \ldots, R. \tag{51}$$

In all experiments, the hierarchy contains $R = 3$ parallel temporal branches operating on the same warped latent-time grid. The branches use kernel sizes

$$\{k_r\}_{r=1}^{3} = \{3, 5, 9\},$$

corresponding to fine-, intermediate-, and coarse-scale temporal receptive fields. Each branch consists of a one-dimensional convolution from C input channels to C hidden channels, followed by a GELU activation and a 1×1 convolution that projects the representation back to C channels. All temporal convolutions use unit stride and symmetric padding $\lfloor k_r/2 \rfloor$, thereby preserving the latent-grid length. No temporal downsampling or dilation is used.

Here, the term "hierarchical" refers to the ordered fine-, intermediate-, and coarse-scale receptive-field levels and their subsequent adaptive spectral–temporal fusion, rather than to separate temporal grids or explicit downsampling stages.

The branch outputs are fused pointwise using input-dependent softmax weights. The gating network is conditioned on the warped latent representation together with a one-channel temporal-geometry context derived from the sampled stochastic warp $\phi^{(k)}$. Consequently, the contribution of each receptive-field scale can vary across latent time locations and across stochastic temporal geometries. The same three-branch configuration is used across all benchmarks.

C.5 Geometry-Conditioned Adaptive Gating

The one-channel temporal-geometry context associated with the k-th sampled warp is defined from the local Jacobian of the physical-to-latent-time transformation. Let

$$s^{(k)}(\tau) = \left(\phi^{(k)}\right)^{-1}(\tau)$$

denote the physical-time location corresponding to latent coordinate τ. We define

$$c_{\mathrm{temp}}^{(k)}(\tau) = \dot{\phi}^{(k)}\left(s^{(k)}(\tau)\right), \qquad c_{\mathrm{temp}}^{(k)}(\tau) \in \mathbb{R}_{>0}. \tag{52}$$

Thus, the temporal context supplied to the gating network directly represents the local amount of latent temporal resolution allocated per unit of physical time.

For the piecewise-linear temporal map used by SHIFT, the derivative is constant on each physical-time interval. Specifically, for $s \in (s_i, s_{i+1})$,

$$\dot{\phi}^{(k)}(s) = \frac{\phi^{(k)}(s_{i+1}) - \phi^{(k)}(s_i)}{s_{i+1} - s_i} = \frac{m_i^{(k)}}{\Delta s_i} = \frac{r_i^{(k)}}{Z_\phi^{(k)}}. \tag{53}$$

Consequently, for $\tau \in [\phi^{(k)}(s_i), \phi^{(k)}(s_{i+1}))$, the corresponding temporal-geometry context is

$$c_{\text{temp}}^{(k)}(\tau) = \frac{r_i^{(k)}}{Z_\phi^{(k)}}. \tag{54}$$

On the discrete latent-time grid, this quantity is evaluated at each grid location according to the warped interval containing that location.

The gating network computes all R branch scores pointwise from the concatenation of the warped latent representation and the corresponding temporal-geometry context:

$$\mathbf{a}^{(k)}(\tau) = G_\eta\left(\left[\tilde{z}^{(k)}(\tau) \parallel c_{\text{temp}}^{(k)}(\tau)\right]\right) \in \mathbb{R}^R, \qquad a_r^{(k)}(\tau) = \left[\mathbf{a}^{(k)}(\tau)\right]_r, \tag{55}$$

where $\parallel$ denotes channel-wise concatenation. Thus, the scale weights depend jointly on the warped signal representation and the local geometry of the sampled stochastic clock.

The normalized gate weight is

$$\alpha_r^{(k)}(\tau) = \frac{\exp\left(a_r^{(k)}(\tau)\right)}{\sum_{\ell=1}^R \exp\left(a_\ell^{(k)}(\tau)\right)}, \qquad \sum_{r=1}^R \alpha_r^{(k)}(\tau) = 1. \tag{56}$$

The adaptively fused multi-scale representation is

$$h_{\text{ms}}^{(k)}(\tau) = \sum_{r=1}^R \alpha_r^{(k)}(\tau) H_r^{(k)}(\tau). \tag{57}$$

C.6 Spectral Refinement and Residual Fusion

The fused multi-scale representation is subsequently refined in the frequency domain. We compute

$$h_{\text{spec}}^{(k)} = P_{\text{spec}}\left[\mathcal{F}^{-1}\left(G_\omega \odot \mathcal{F}\left(h_{\text{ms}}^{(k)}\right)\right)\right], \tag{58}$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ denote the real Fourier transform and its inverse, G_ω is a learnable frequency-dependent modulation shared across all sampled temporal geometries, and P_{spec} is a learned 1×1 channel projection. Although the spectral modulation parameters are shared, the refined representation remains sample-dependent because $h_{\text{ms}}^{(k)}$ depends on the k-th sampled temporal geometry. The refined representation is

$$h_{\text{ref}}^{(k)} = P_{\text{fuse}}\left(h_{\text{ms}}^{(k)} + h_{\text{spec}}^{(k)}\right), \tag{59}$$

Let $h_{\text{in}}^{(k)}$ denote the unrefined representation supplied to the residual-fusion path, with the same channel dimension as $h_{\text{ref}}^{(k)}$. The final output is obtained through learnable residual fusion:

$$h_{\text{out}}^{(k)} = h_{\text{in}}^{(k)} + \beta\left(h_{\text{ref}}^{(k)} - h_{\text{in}}^{(k)}\right), \qquad 0 \le \beta \le 1. \tag{60}$$

This residual parameterization allows the spectral–temporal module to start close to the original operator representation and progressively increase its contribution during optimization.

The adaptive weights in equation 56 allow SHIFT to emphasize the temporal scale most relevant to the latent sequence and the sampled temporal geometry.

C.7 Prediction Head

For the k-th sampled temporal geometry, the hierarchical spectral–temporal module produces the refined representation $h_{\text{out}}^{(k)}$. The forecasting decoder maps this representation to the conditional predictive mean:

$$\hat{y}^{(k)}(\tau) = D_\omega\left(h_{\text{out}}^{(k)}(\tau), \tau\right). \tag{61}$$

A separate pointwise variance head predicts the corresponding conditional log-variance:

$$\ell_{v,k}(\tau) = G_\nu\left(h_{\text{out}}^{(k)}(\tau)\right), \tag{62}$$

where G_ν is implemented as a learned 1×1 channel projection. The positive conditional variance is obtained as

$$\sigma_{v,k}^2(\tau) = \exp\left(\ell_{v,k}(\tau)\right) + \epsilon_0, \qquad \epsilon_0 > 0, \tag{63}$$

with numerical bounding applied when required for stable optimization.

These outputs define the conditional Gaussian predictive model

$$p_\omega\left(y(\tau) \mid x, \phi^{(k)}\right) = \mathcal{N}\left(y(\tau); \hat{y}^{(k)}(\tau), \sigma_{v,k}^2(\tau)\right). \tag{64}$$

For spatio-temporal forecasting, the same heads are applied pointwise over spatial locations and output channels:

$$\begin{aligned} \hat{u}^{(k)}(a,\tau) &= D_\omega\left(h_{\text{out}}^{(k)}(a,\tau), a, \tau\right), \\ \sigma_{v,k}^2(a,\tau) &= \exp\left[G_\nu\left(h_{\text{out}}^{(k)}(a,\tau)\right)\right] + \epsilon_0. \end{aligned} \tag{65}$$

The quantity $\sigma_{v,k}^2$ represents the conditional predictive-variance component under the k-th sampled temporal geometry. We use its Monte Carlo average as the estimate of aleatoric uncertainty U_{alea}. This terminology does not imply that the estimate isolates irreducible observation noise, because the learned conditional variance may also absorb model mismatch and other unrepresented sources of predictive uncertainty.

C.8 Predictive Mixture and Uncertainty Decomposition

SHIFT treats temporal geometry as a random variable conditioned on the observed history. During inference, the model draws K stochastic temporal geometries:

$$\phi^{(k)} \sim P_\psi\left(\phi \mid x\right), \qquad k = 1, \ldots, K. \tag{66}$$

Using the conditionally diagonal Gaussian predictive distribution defined in Appendix C.7, the K sampled temporal geometries define a finite Monte Carlo predictive mixture over the complete forecast vector:

$$p_K\left(\mathbf{y} \mid x\right) = \frac{1}{K}\sum_{k=1}^{K} \mathcal{N}\left(\mathbf{y}; \hat{\mathbf{y}}^{(k)}, \operatorname{diag}\left(\boldsymbol{\sigma}_{v,k}^2\right)\right). \tag{67}$$

The corresponding marginal predictive distribution for scalar forecast component d is

$$p_K\left(y_d \mid x\right) = \frac{1}{K}\sum_{k=1}^{K} \mathcal{N}\left(y_d; \hat{y}_d^{(k)}, \sigma_{v,k,d}^2\right). \tag{68}$$

The corresponding Monte Carlo predictive mean is

$$\bar{y}_d = \frac{1}{K}\sum_{k=1}^{K} \hat{y}_d^{(k)}. \tag{69}$$

Uncertainty in the predictive mixture arises from two modeled sources. The law of total variance gives

$$\operatorname{Var}\left(Y_d \mid x\right) = \mathbb{E}_{\phi \mid x}\left[\operatorname{Var}\left(Y_d \mid x, \phi\right)\right] + \operatorname{Var}_{\phi \mid x}\left[\mathbb{E}\left(Y_d \mid x, \phi\right)\right]. \tag{70}$$

The conditional value-uncertainty component is estimated as

$$U_{\text{alea},d} = \frac{1}{K}\sum_{k=1}^{K} \sigma_{v,k,d}^2. \tag{71}$$

We retain the notation U_{alea} as shorthand for the conditional predictive-variance component. It should not be interpreted as isolating irreducible observation noise, because the learned conditional variance may also absorb model mismatch and other unrepresented sources of predictive uncertainty.

The temporal-geometry-induced component is

$$U_{\text{temp},d} = \frac{1}{K} \sum_{k=1}^{K} \left(\hat{y}_d^{(k)} - \bar{y}_d \right)^2. \tag{72}$$

Because different sampled warpings change how temporal regions are expanded, compressed, aligned, and represented on the latent-time grid, U_{temp} measures predictive variability induced by the model's stochastic temporal reparameterization.

The total predictive uncertainty represented by SHIFT is

$$U_{\text{total},d} = U_{\text{alea},d} + U_{\text{temp},d}. \tag{73}$$

Equivalently, the variance of the finite predictive mixture is

$$\text{Var}_{p_K}\left(Y_d \mid x\right) = \frac{1}{K} \sum_{k=1}^{K} \sigma_{v,k,d}^2 + \frac{1}{K} \sum_{k=1}^{K} \left(\hat{y}_d^{(k)} - \bar{y}_d \right)^2. \tag{74}$$

The normalization $1/K$, rather than $1/(K-1)$, is used because U_{temp} is the variance of the finite equally weighted predictive mixture represented by the sampled components, rather than an unbiased estimator of an unknown population variance.

The decomposition is evaluated element-wise under the conditional diagonal-covariance decoder. Consequently, the scalar uncertainty summaries reported in the experiments do not represent cross-variable, cross-horizon, or cross-spatial covariance terms.

C.9 Inference

At inference time, SHIFT draws K temporal geometries from the learned conditional warp distribution using the stochastic construction defined in Appendix C.2. For each sampled geometry, the encoded observations are interpolated onto the uniform latent-time grid described in Appendix C.3.

The resulting latent representation is processed by the geometry-conditioned multi-scale spectral–temporal module and decoded into a conditional predictive mean and variance. The K conditional predictive distributions are then aggregated according to Appendix C.8 to obtain the predictive mean, conditional value uncertainty, temporal-geometry-induced uncertainty, and total predictive uncertainty.

Thus, inference differs across samples only through the sampled temporal geometry and its resulting latent representation; the forecasting network parameters are shared across all samples.

C.10 Interpretation and Identifiability

The induced measure $d\mu_\phi(s) = \dot{\phi}(s)\, ds$ is completely determined by the sampled temporal reparameterization. It is not an independently identifiable importance measure. In an unrestricted operator, changes in the map, kernel, and latent representation may compensate for one another. We therefore interpret the learned map as an architectural parameterization whose utility must be established empirically, rather than as recovery of a unique physical temporal geometry.

A larger value of $\dot{\phi}(s)$ assigns a larger portion of the normalized latent-time grid to the corresponding physical-time region. This does not by itself prove that the region is intrinsically more important; its predictive effect depends jointly on the interpolation, kernel, multi-scale branches, and decoder.

D OBJECTIVE AND REGULARIZATION

Let D denote the number of scalar forecast targets in an evaluated sequence, including all forecast times and output dimensions. The mean-squared forecasting loss is

$$\mathcal{L}_{\mathrm{MSE}} = \frac{1}{D}\sum_{d=1}^{D}\left(y_d - \bar{y}_d\right)^2, \tag{75}$$

where $\bar{y}$ is the Monte Carlo predictive mean.

When the conditional variance head is trained, we use the average conditional Gaussian negative log-likelihood over the sampled temporal geometries, up to the additive constant $\frac{1}{2}\log(2\pi)$:

$$\mathcal{L}_{\mathrm{NLL}} = \frac{1}{KD}\sum_{k=1}^{K}\sum_{d=1}^{D}\left[\frac{\left(y_d - \hat{y}_d^{(k)}\right)^2}{2\sigma_{v,k,d}^2} + \frac{1}{2}\log\left(\sigma_{v,k,d}^2\right)\right]. \tag{76}$$

This objective averages the conditional Gaussian negative log-likelihoods associated with the sampled temporal geometries. It is not the negative log-likelihood of the finite Monte Carlo predictive mixture in equation 67. For the objective-sensitivity analysis, we additionally consider the negative log-likelihood of the finite K-component Monte Carlo mixture, normalized per scalar forecast target. For a sampled temporal geometry k, define the joint diagonal-Gaussian log likelihood over the D scalar forecast targets as

$$\ell_k(y) = \sum_{d=1}^{D}\log\mathcal{N}\left(y_d; \hat{y}_d^{(k)}, \sigma_{v,k,d}^2\right). \tag{77}$$

The corresponding finite-mixture negative log-likelihood is

$$\mathcal{L}_{\mathrm{mix}} = -\frac{1}{D}\left[\log\left(\sum_{k=1}^{K}\exp(\ell_k(y))\right) - \log K\right]. \tag{78}$$

In implementation, the component aggregation is evaluated using a numerically stable log-sum-exp operation. The log-sum-exp term corresponds to the exact likelihood of the represented finite K-component Monte Carlo mixture, while the factor $1/D$ reports the negative log-likelihood per scalar forecast target. This quantity should not be interpreted as the exact marginal likelihood under the continuous conditional distribution over temporal geometries. We use this finite-mixture NLL only as a controlled sensitivity alternative to the component-wise conditional objective. Let $\Phi_i^{(k)}$ denote the cumulative temporal coordinate before endpoint normalization for the k-th sampled temporal geometry:

$$\Phi_1^{(k)} = 0, \qquad \Phi_{i+1}^{(k)} = \Phi_i^{(k)} + r_i^{(k)}\Delta s_i, \qquad i = 1, \ldots, N-1. \tag{79}$$

The monotone temporal map used by the neural operator is obtained by normalizing this cumulative coordinate:

$$\phi_i^{(k)} = \frac{\Phi_i^{(k)} - \Phi_1^{(k)}}{\Phi_N^{(k)} - \Phi_1^{(k)}}. \tag{80}$$

Because the positive interval rates and observation intervals imply $\Phi_N^{(k)} - \Phi_1^{(k)} > 0$, the normalization is well defined and satisfies $\phi_1^{(k)} = 0$ and $\phi_N^{(k)} = 1$ exactly. In implementation, positivity of the interval rates is numerically safeguarded before the cumulative normalization; no additive constant is included in the normalization denominator. A smoothness penalty is applied to the pre-normalization cumulative temporal coordinate to discourage abrupt variation between adjacent temporal increments:

$$\mathcal{L}_{\mathrm{smooth}} = \frac{1}{K(N-2)}\sum_{k=1}^{K}\sum_{i=2}^{N-1}\left(\Phi_{i+1}^{(k)} - 2\Phi_i^{(k)} + \Phi_{i-1}^{(k)}\right)^2. \tag{81}$$

Using

$$\Phi_{i+1}^{(k)} - \Phi_i^{(k)} = r_i^{(k)}\Delta s_i,$$

the regularizer can equivalently be viewed as penalizing variation between consecutive pre-normalization temporal increments. On irregular physical-time grids, it should therefore be interpreted as an increment-smoothness penalty rather than as a finite-difference approximation to physical-time curvature $d^2\phi/ds^2$. The penalty discourages abrupt changes in the pre-normalization allocation of latent temporal mass while preserving monotonicity and input-dependent temporal expansion and compression.

A log-barrier regularizer is applied to the positive pre-normalization interval rates:

$$\mathcal{L}_{\text{rate}} = -\frac{1}{K(N-1)} \sum_{k=1}^{K} \sum_{i=1}^{N-1} \log\left(r_i^{(k)} + \epsilon_0\right). \tag{82}$$

Because $r_i^{(k)}$ is the positive pre-normalization interval rate, this term discourages very small absolute rate values. It should not be interpreted as directly enforcing a lower bound on the normalized interval masses

$$m_i^{(k)} = \frac{r_i^{(k)} \Delta s_i}{\sum_{j=1}^{N-1} r_j^{(k)} \Delta s_j}, \tag{83}$$

which depend on the relative rates and observation intervals. An optional length-consistency constraint controls the global span of the cumulative temporal coordinate before normalization:

$$\mathcal{L}_{\text{length}} = \frac{1}{K} \sum_{k=1}^{K} \left[\left(\Phi_N^{(k)} - \Phi_1^{(k)}\right) - (s_N - s_1)\right]^2. \tag{84}$$

The normalized temporal map is invariant to a common positive rescaling of all interval rates, but the pre-normalization regularizers above are not. Consequently, these terms can select an absolute rate scale during optimization even though that scale is not identifiable from the normalized temporal geometry alone. The distinction is between scale invariance of the forecasting map after normalization and scale dependence introduced by the training regularization. When variational regularization is active, the reference distribution over the pre-softplus interval variables is

$$p_0(a_{1:N-1}) = \prod_{i=1}^{N-1} \mathcal{N}(a_i; 0, 1). \tag{85}$$

The corresponding KL regularizer is

$$\begin{aligned} \mathcal{L}_{\text{KL}} &= \text{KL}\left(p_\psi(a_{1:N-1} \mid x) \,\|\, p_0(a_{1:N-1})\right) \\ &= \frac{1}{2} \sum_{i=1}^{N-1} \left[\mu_{\phi,i}^2 + \sigma_{\phi,i}^2 - \log\left(\sigma_{\phi,i}^2\right) - 1\right]. \end{aligned} \tag{86}$$

E Experimental Protocol

We evaluate SHIFT across irregular, regular, partially observed, and spatio-temporal forecasting benchmarks. This appendix describes the datasets, preprocessing and data partitions, comparison baselines, evaluation metrics, optimization settings, and computational resources used in the experiments.

E.1 Datasets and Tasks

We evaluate SHIFT on the same family of forecasting benchmarks used in the KRNO time-shift operator study (Dama et al., 2025). The benchmark suite includes irregularly sampled clinical, climate, human-motion, and simulated dynamical-system datasets, as well as regularly sampled financial, competition, and sports-trajectory datasets. We also consider a spatio-temporal shallow-water forecasting problem. This allows us to evaluate whether the proposed stochastic temporal geometry is beneficial across different sampling regimes and data modalities.

Irregularly sampled time series. We use three irregular multivariate time-series benchmarks: MIMIC-III, USHCN, and Human Activity. MIMIC-III is a credentialed-access critical-care database containing electronic health records from intensive-care-unit patients (Johnson et al., 2016). Following the preprocessing protocol used in KRNO, each time series contains 96 variables recorded during the first 48 hours after admission, and the task is to forecast the next 24-hour period from the first 24 hours of observations. The dataset is available through PhysioNet.[1]

USHCN is a climate dataset based on the United States Historical Climatology Network (Menne et al., 2015). It contains daily measurements of climate variables such as precipitation, snowfall, snow depth, and minimum and maximum temperature. Following the KRNO protocol, we use the preprocessed subset with 1,114 stations over the period 1996–2000, resulting in 26,736 multivariate time series. The task is to forecast the next 24 months from the previous 24 months of climate observations. The dataset source is provided by NOAA/NCEI.[2]

The Human Activity dataset contains 3D positional measurements from four sensors worn on the left ankle, right ankle, belt, and chest. Following the preprocessing used in prior irregular time-series benchmarks (Rubanova et al., 2019; Zhang et al., 2024), the data consist of 5,400 irregularly sampled multivariate time series. Each sequence spans 4,000 milliseconds, where the first 3,000 milliseconds are used as observations and the next 1,000 milliseconds are used as the forecasting target. The original dataset is available from the UCI Machine Learning Repository.[3]

MuJoCo. We evaluate SHIFT on the MuJoCo Hopper benchmark, following the setup used in KRNO and prior continuous-time forecasting work (Tassa et al., 2018; Rubanova et al., 2019). Each trajectory is a 14-dimensional state sequence sampled at 100 evenly spaced time points. The model predicts the next 10 time steps using the preceding 50 time steps. To evaluate robustness to irregular observations, we follow the KRNO protocol and randomly remove 30%, 50%, or 70% of observations from each trajectory. The simulation environment is based on the DeepMind Control Suite.[4]

Regularly sampled time series. For regularly sampled forecasting, we use M4, Crypto, and Player Trajectory. M4 is a large-scale univariate forecasting benchmark containing 100,000 time series from multiple domains and seasonalities (Makridakis et al., 2020). Following the KRNO comparison, we evaluate on the M4 subsets used in that benchmark and report sMAPE. The dataset is available through the M4 competition resources.[5]

The Crypto dataset is a multivariate financial forecasting benchmark containing trading-related features for 14 cryptocurrencies. Following the KRNO and KNF protocols, the task is to forecast future cryptocurrency returns and evaluate performance using weighted RMSE (Ticchi et al., 2021; Wang et al., 2023). The data source follows the G-Research Crypto Forecasting benchmark.[6]

The Player Trajectory dataset contains two-dimensional basketball player positions, and the forecasting task is to predict future player movement trajectories. Following KRNO, we use the same train, validation, and test splits as in Wang et al. (2023) and evaluate using RMSE. The raw player-movement data are based on NBA SportVU logs.[7]

Spatio-temporal forecasting. We also evaluate SHIFT on the shallow-water benchmark used in KRNO. The dataset is taken from the LOCA benchmark of Kissas et al. (2022). It consists of simulated shallow-water dynamics on a 32×32 spatial grid over the time interval $(0, 1)$, sampled every 0.01 seconds. The task is to predict three physical fields: fluid height ρ, horizontal velocity u, and vertical velocity v. Following the KRNO setup, both the look-back and prediction windows are

[1] https://physionet.org/content/mimiciii/1.4/

[2] https://www.ncei.noaa.gov/products/land-based-station/us-historical-climatology-network

[3] https://archive.ics.uci.edu/dataset/196/localization+data+for+person+activity

[4] https://github.com/google-deepmind/dm_control

[5] https://github.com/Mcompetitions/M4-methods

[6] https://www.kaggle.com/competitions/g-research-crypto-forecasting/data

[7] https://github.com/linouk23/NBA-Player-Movements

set to 0.05 seconds, and predictions are recursively rolled out until 0.6 seconds. The code and data are provided by the LOCA repository.[8]

Overall, these datasets cover clinical monitoring, climate observations, human motion, simulated physical dynamics, financial markets, sports trajectories, and spatio-temporal fluid dynamics. This diversity allows us to test whether SHIFT improves forecasting not only under irregular sampling and missing observations, but also under regular sampling where temporal regions may have non-uniform predictive importance.

E.2 Preprocessing and Data Splits

SHIFT retains the benchmark-specific preprocessing and evaluation pipelines used in the corresponding KRNO experiments. Unless otherwise stated, we use the same processed inputs, train/validation/test partitions, observation and forecast windows, normalization procedures, and evaluation metrics. The stochastic temporal reparameterization is applied only after the benchmark-specific input representation has been constructed. Here, "benchmark protocol" refers to the dataset construction, preprocessing, train/validation/test partitions, observation and forecast windows, normalization conventions, and evaluation metrics inherited from the corresponding KRNO experiment. The SHIFT-specific architecture, stochastic temporal reparameterization, probabilistic training terms, and temporal-geometry regularization are reported separately below and are not implied to be inherited from KRNO.

For MIMIC-III, USHCN, and Human Activity, we retain the irregular time-series preprocessing, observation masks, and data partitions used by the T-PatchGNN-based pipeline adopted by KRNO. The processed irregular inputs are supplied to SHIFT without introducing an additional imputation or mask-construction procedure.

For MuJoCo, we retain the data-generation, train/test partitioning, observation-removal, and evaluation protocol used in the corresponding Stable Neural SDE benchmark. We evaluate the regularly observed setting and the variants in which 30%, 50%, or 70% of input observations are randomly removed.

For M4, Crypto, and Player Trajectory, we retain the regular time-series windowing and data partitions used in the corresponding KRNO experiments. For each M4 training series, the final 10% of the available training portion is used for validation. For Crypto and Player Trajectory, the final 10% of each training series is reserved for validation, with prediction horizons of 15 and 30 steps, respectively. Following the KRNO benchmark protocol (Dama et al., 2025) and the data splits of Wang et al. (2023), evaluation is performed on the complete Player Trajectory test split. All test windows are included when computing RMSE over the 30-step prediction horizon; no test-window subsampling or cap on the number of evaluation batches is applied.

For Shallow Water, we use the same LOCA-derived spatial grid, look-back window, forecast window, and recursive rollout protocol as in the corresponding KRNO experiment. SHIFT modifies only the temporal forecasting architecture while retaining the original spatial discretization and target construction.

E.3 Baselines and Reporting Protocol

We compare SHIFT primarily against KRNO (Dama et al., 2025) because the proposed framework retains a KRNO-derived operator backbone. However, the complete SHIFT configuration also introduces input-conditioned stochastic temporal reparameterization, latent-grid resampling, hierarchical multi-scale spectral–temporal processing, adaptive gating, and a probabilistic prediction head. Therefore, the full-model comparison with KRNO evaluates the complete SHIFT architecture rather than isolating the effect of temporal reparameterization alone. Component-specific attribution is made through the controlled ablation experiments reported in Section 3.2.

For the irregular multivariate comparison in Table 1, we include DLinear (Zeng et al., 2023), TimesNet (Wu et al., 2023), PatchTST (Nie et al., 2023), Crossformer (Zhang & Yan, 2023), MTGNN (Wu et al., 2020), StemGNN (Cao et al., 2020), CrossGNN (Huang et al., 2023), FourierGNN (Yi et al., 2023), SeFT (Horn et al., 2020), RainDrop (Zhang et al., 2022), Warpformer (Zhang et al.,

[8] https://github.com/PredictiveIntelligenceLab/LOCA

2023), mTAND (Shukla & Marlin, 2021), Latent ODE (Rubanova et al., 2019), CRU (Schirmer et al., 2022), Neural Flow (Biloš et al., 2021), and T-PatchGNN (Zhang et al., 2024). These baselines cover forecasting models based on linear, transformer, graph, set-function, and continuous-time representations. Baseline values through T-PatchGNN follow Zhang et al. (2024), as reproduced in Dama et al. (2025), while KRNO is rerun under our corresponding benchmark configuration.

For the MuJoCo robustness experiments, we additionally compare against continuous-time and stochastic-dynamics baselines. These include ACE-NODE (Jhin et al., 2021), Neural CDE (NCDE) (Kidger et al., 2020), Attentive Neural CDE (ANCDE) (Jhin et al., 2024), EXIT (Jhin et al., 2022), LEAP (Jhin et al., 2023), and Neural SDE (Li et al., 2020). The GRU-Δt, Neural LSDE, Neural LNSDE, and Neural GSDE results follow Oh et al. (2024). KRNO results are obtained from our reruns under the corresponding benchmark protocol.

For regularly sampled benchmarks, we compare against the corresponding baselines used in the KRNO evaluation. For M4, these include FFORMA, Smyl, N-BEATS, and KNF (Montero-Manso et al., 2020; Smyl, 2020; Oreshkin et al., 2020; Wang et al., 2023). For Crypto and Player Trajectory, we compare against KNF, FEDformer, LEM, VBS, MLP-based models, random-forest baselines, and VARIMA (Wang et al., 2023; Zhou et al., 2022; Rusch et al., 2022; Li et al., 2021a; Stock & Watson, 2001). We follow the KRNO benchmark protocol, which in turn uses the data splits and testing-window lengths of Wang et al. (2023). Accordingly, the previously reported baseline results are compared under this shared benchmark protocol, while KRNO is rerun directly alongside SHIFT.

For the shallow-water spatio-temporal forecasting task, we compare SHIFT against KRNO and the neural-operator baselines reported in the KRNO study, including FNO-3D and LOCA (Li et al., 2021b; Kissas et al., 2022). This comparison evaluates whether the proposed stochastic temporal geometry remains useful when the forecasting problem involves both spatial and temporal structure.

The purpose of these comparisons is twofold. First, we evaluate whether SHIFT improves over the direct KRNO baseline. Second, we test whether the proposed temporally reparameterized formulation is competitive with specialized forecasting baselines across different data regimes. Whenever possible, we use the same train/validation/test splits, forecasting horizons, preprocessing procedures, and evaluation metrics as those used in the KRNO benchmark.

Reporting conventions. We compare SHIFT with KRNO and with previously reported forecasting approaches under the corresponding benchmark protocols. The KRNO and SHIFT results reported in this work are obtained from our own experiments using the corresponding benchmark configurations. Unless otherwise stated, results for the remaining non-SHIFT baselines are taken from the cited benchmark studies.

KRNO is rerun alongside SHIFT using the corresponding benchmark protocols. For other baselines whose values are taken from prior work, we use the same datasets, forecasting horizons, preprocessing conventions, and evaluation metrics whenever these details are available. Comparisons with those externally reported baselines should therefore be interpreted separately from the directly rerun KRNO comparison. For SHIFT and the directly rerun KRNO baseline, we follow the corresponding benchmark-specific evaluation protocol. Results are reported as mean $\pm$ standard deviation when available; statistics for externally reported baselines are reproduced from the cited benchmark studies.

Bold, underlined, and italic entries denote the first-, second-, and third-best mean values among the methods shown, respectively. Equal reported means share the same rank. Lower values indicate better performance for all point-forecasting metrics. Uncertainty magnitudes are reported separately and are not interpreted as lower-is-better metrics.

E.4 Evaluation Metrics

We evaluate SHIFT using the forecasting metrics adopted by the corresponding benchmark protocols. Point-forecasting errors are reported such that lower values indicate better predictive accuracy. For probabilistic forecasts, we additionally evaluate the quality of the estimated predictive uncertainty using negative log-likelihood, empirical prediction-interval coverage, coverage gap, and calibration error.

For MIMIC-III, USHCN, and Human Activity, we report mean squared error (MSE) and mean absolute error (MAE), following the irregular time-series evaluation protocol used in KRNO and T-PatchGNN (Dama et al., 2025; Zhang et al., 2024). For MuJoCo, we report MSE under fully observed sequences and under 30%, 50%, and 70% randomly dropped observations. For the Crypto benchmark, we use weighted root mean squared error (weighted RMSE), while standard RMSE is used for Player Trajectory. For the M4 subsets, we report symmetric mean absolute percentage error (sMAPE). For shallow-water forecasting, we report the relative L_2 error separately for the fluid-height field ρ and the velocity fields u and v.

Let N denote the total number of evaluated forecast targets, y_i the ground-truth value, and $\hat{y}_i$ the corresponding point prediction. The mean squared error is defined as

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 . \tag{87}$$

The mean absolute error is

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| . \tag{88}$$

The root mean squared error is computed as

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}. \tag{89}$$

For benchmarks with target-specific weights, the weighted RMSE is defined as

$$\mathrm{WRMSE} = \sqrt{\frac{\sum_{i=1}^{N} w_i\,(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N} w_i}}, \tag{90}$$

where w_i denotes the benchmark-provided weight associated with the i-th forecast target.

For the M4 subsets, the symmetric mean absolute percentage error is computed as

$$\mathrm{sMAPE} = \frac{200}{N}\sum_{i=1}^{N}\frac{|y_i - \hat{y}_i|}{|y_i| + |\hat{y}_i| + \epsilon_0}, \tag{91}$$

where $\epsilon_0 > 0$ is a small constant used for numerical stability.

For the shallow-water benchmark, the relative L_2 error of a predicted field $\hat{u}$ is

$$\epsilon_u = \frac{\|u - \hat{u}\|_2}{\|u\|_2 + \epsilon_0}. \tag{92}$$

Predictive uncertainty. The predictive mean and the conditional value, temporal-geometry-induced, and total predictive uncertainty components are defined in Appendix C.8. For probabilistic evaluation, however, we retain the individual conditional Gaussian components associated with the sampled temporal geometries and evaluate the resulting finite Monte Carlo predictive mixture directly.

Finite-Mixture Negative Log-Likelihood. For the i-th scalar forecast target, the K sampled temporal geometries define the finite predictive mixture

$$p_{K,i}(y \mid x_i) = \frac{1}{K}\sum_{k=1}^{K}\mathcal{N}\left(y; \hat{y}_i^{(k)}, \sigma_{v,k,i}^2\right). \tag{93}$$

We evaluate this predictive distribution directly using

$$\mathrm{NLL}_{\mathrm{mix}} = -\frac{1}{N}\sum_{i=1}^{N}\log\left[\frac{1}{K}\sum_{k=1}^{K}\mathcal{N}\left(y_i; \hat{y}_i^{(k)}, \sigma_{v,k,i}^2\right)\right]. \tag{94}$$

The mixture likelihood is evaluated using a numerically stable log-sum-exp computation. Lower values indicate that the represented finite predictive mixture assigns higher probability density to the observed targets.

Gaussian-Mixture Continuous Ranked Probability Score. For the i-th scalar target, the cumulative distribution function of the finite predictive mixture is

$$F_i^{\text{mix}}(z) = \frac{1}{K}\sum_{k=1}^{K} \Phi\left(\frac{z - \hat{y}_i^{(k)}}{\sigma_{v,k,i}}\right), \tag{95}$$

where Φ denotes the standard normal cumulative distribution function.

The continuous ranked probability score is evaluated for this mixture distribution as

$$\text{CRPS}_{\text{mix}} = \frac{1}{N}\sum_{i=1}^{N} \int_{-\infty}^{\infty} \left[F_i^{\text{mix}}(z) - \mathbb{I}\left(y_i \le z\right)\right]^2 dz. \tag{96}$$

Lower values indicate better agreement between the observed targets and the finite predictive mixture.

Finite-Mixture Interval Score. For a nominal central coverage level α, let $q_{i,p}^{\text{mix}}$ denote the p-quantile of the finite predictive mixture, defined by

$$F_i^{\text{mix}}\left(q_{i,p}^{\text{mix}}\right) = p. \tag{97}$$

The corresponding central prediction interval is

$$\left[l_i^{(\alpha)}, u_i^{(\alpha)}\right] = \left[q_{i,(1-\alpha)/2}^{\text{mix}}, q_{i,(1+\alpha)/2}^{\text{mix}}\right]. \tag{98}$$

The quantiles are obtained numerically from the finite-mixture CDF in Equation 95. The interval score is

$$\begin{aligned} \text{IS}_\alpha = \frac{1}{N}\sum_{i=1}^{N} \Bigg[& u_i^{(\alpha)} - l_i^{(\alpha)} \\ & + \frac{2}{1-\alpha}\left(l_i^{(\alpha)} - y_i\right)\mathbb{I}\left[y_i < l_i^{(\alpha)}\right] \\ & + \frac{2}{1-\alpha}\left(y_i - u_i^{(\alpha)}\right)\mathbb{I}\left[y_i > u_i^{(\alpha)}\right]\Bigg]. \end{aligned} \tag{99}$$

Lower values indicate narrower predictive intervals subject to penalties for observations falling outside the interval.

Prediction-interval coverage. For a nominal central coverage level α, we use the finite-mixture quantiles defined in Equation 97. The central prediction interval with nominal coverage level α is therefore

$$\mathcal{I}_i^{(\alpha)} = \left[q_{i,(1-\alpha)/2}^{\text{mix}}, q_{i,(1+\alpha)/2}^{\text{mix}}\right]. \tag{100}$$

For the 95% prediction intervals reported in the main text, $\alpha = 0.95$, giving

$$\mathcal{I}_i^{(0.95)} = \left[q_{i,0.025}^{\text{mix}}, q_{i,0.975}^{\text{mix}}\right]. \tag{101}$$

The empirical prediction-interval coverage at nominal level α is computed as

$$\text{Coverage}_\alpha = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}\left[y_i \in \mathcal{I}_i^{(\alpha)}\right], \tag{102}$$

where $\mathbb{I}[\cdot]$ denotes the indicator function. A calibrated predictive distribution should yield empirical coverage close to the corresponding nominal level.

For the 95% intervals used in our reported results, we additionally report the absolute coverage gap

$$\text{Gap}_{95} = \left|\text{Coverage}_{0.95} - 0.95\right|. \tag{103}$$

Smaller values indicate closer agreement between empirical and nominal coverage.

Calibration error. At each nominal level, empirical coverage is computed using the corresponding central quantiles of the finite predictive mixture. Coverage at a single nominal level does not fully characterize predictive calibration. We therefore evaluate calibration over a collection of nominal central coverage levels

$$\mathcal{A} = \{0.50,\ 0.60,\ 0.70,\ 0.80,\ 0.90,\ 0.95\}\,.$$

Here, $J = 6$ denotes the number of nominal coverage levels. For each $\alpha_j \in \mathcal{A}$, the corresponding central interval is constructed directly from the finite-mixture quantiles as

$$\mathcal{I}_i^{(\alpha_j)} = \left[q^{\mathrm{mix}}_{i,(1-\alpha_j)/2}, q^{\mathrm{mix}}_{i,(1+\alpha_j)/2}\right], \tag{104}$$

The empirical coverage at nominal level α_j is

$$\widehat{C}(\alpha_j) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}\left[y_i \in \mathcal{I}_i^{(\alpha_j)}\right]. \tag{105}$$

We summarize the deviation between nominal and empirical coverage using the mean absolute calibration error

$$\mathrm{CalErr} = \frac{1}{J}\sum_{j=1}^{J}\left|\widehat{C}(\alpha_j) - \alpha_j\right|. \tag{106}$$

Lower calibration error indicates closer agreement between the nominal coverage probabilities and the empirical frequencies with which the observed targets fall inside the corresponding finite-mixture prediction intervals. Importantly, all intervals used for coverage and calibration are obtained directly from the predictive Gaussian mixture induced by the sampled temporal geometries, rather than from a moment-matched Gaussian approximation.

E.5 Hyperparameters and Implementation Details

SHIFT modules. For the temporal forecasting configurations, the local temporal context encoder uses latent dimension 32, while the probabilistic time-warp module uses hidden dimension 64. Let $d_C = \dim(C_i)$ denote the dimension of the interval-level conditioning context defined in Equation 30. The conditional-mean head $g_{\mu,\psi}$ and the log-variance head $g_{\sigma,\psi}$ each use

$$\mathrm{Linear}(d_C, 64) \to \mathrm{GELU} \to \mathrm{Linear}(64, 1).$$

Their outputs define

$$\mu_{\phi,i} = g_{\mu,\psi}(C_i), \qquad \ell_{\phi,i} = g_{\sigma,\psi}(C_i), \qquad \sigma^2_{\phi,i} = \exp(\ell_{\phi,i}) + \epsilon_0,$$

consistent with Equations 32–34. A latent interval variable is then sampled using

$$a_i = \mu_{\phi,i} + \sigma_{\phi,i}\xi_i, \qquad \xi_i \sim \mathcal{N}(0,1),$$

and converted to a positive interval rate through

$$r_i = \mathrm{softplus}(a_i).$$

The resulting positive rates are normalized to construct the monotone temporal map as described in Appendix C.2.

The dedicated Shallow Water temporal module instead uses hidden dimension 32. No additional custom initialization is applied to the Linear layers of the probabilistic warp heads.

Residual-fusion parameterization. For temporal configurations using adaptive residual fusion, the coefficient in Equation 60 is implemented as

$$\beta = \sigma(b_\beta)\, s_{\mathrm{ext}}, \qquad 0 \le s_{\mathrm{ext}} \le 1,$$

where b_β is learnable and initialized to -2.4. The external strength s_{ext} is controlled by the training schedule. At unit external strength, the initialization corresponds to $\sigma(-2.4) \simeq 0.083$.

Latent-grid resolution. The symbol M in the latent-time formulation denotes the number of points at which the temporal map is represented. In the reported implementations, this resolution follows the active temporal/operator grid and is not independently tuned as an additional SHIFT hyperparameter. In the dedicated Shallow Water implementation, the normalized temporal grid contains L points for a sequence of length L.

Numerical safeguards. The analytical notation uses $\epsilon_0 > 0$ as a generic numerical safeguard, whereas the implementation uses operation-specific constants. Temporal-gap and positive-rate safeguards use 10^{-6} in the temporal forecasting implementations, while normalized-coordinate and interpolation denominators use 10^{-8}. The dedicated Shallow Water temporal warp uses a 10^{-4} positive-rate floor and 10^{-8} for normalization and interpolation denominators. These constants are numerical safeguards rather than optimized hyperparameters.

Optimization. The temporal forecasting models are optimized primarily using AdamW. A ReduceLROnPlateau scheduler is employed for the irregular time-series experiments, and early stopping is applied according to the validation forecasting loss. Gradient accumulation is used for experiments with a physical batch size of one. Model selection is based exclusively on validation performance, after which the selected checkpoint is evaluated on the test set.

The shallow-water model is optimized using Adam with a cosine annealing learning-rate schedule. Its KRNO backbone contains five kernel integral layers with 48 channels, lifting and projection dimensions of 256, and kernel networks with five hidden layers of 96 units. The model is trained for at most 80 epochs using a batch size of four and gradient clipping at 1.0.

Dataset-specific objectives and regularization. The benchmark data and evaluation protocols follow the corresponding KRNO configurations, including the dataset construction, preprocessing, train/validation/test partitions, observation and forecast windows, normalization conventions, and evaluation metrics. The optimization terms listed below are specific to SHIFT and supplement, rather than replace, these inherited benchmark protocols.

Table 9 reports the forecasting objectives and temporal-geometry regularization coefficients used by the reported SHIFT configurations. We use $\mathcal{L}^{(D)}_{\mathrm{base}}$ to denote the benchmark-specific point-forecasting loss of the corresponding dataset runner, before the addition of SHIFT-specific probabilistic and temporal-geometry terms. Thus, $\mathcal{L}^{(D)}_{\mathrm{base}}$ is dataset-dependent rather than a single loss shared across benchmarks. The exact base loss used for each dataset is reported together with the corresponding implementation settings below. The implementation variable corresponding to the log-rate barrier $\lambda_{\mathrm{rate}}\mathcal{L}_{\mathrm{rate}}$ is named `lambda_entropy` in the temporal-model code.

For MIMIC-III, the corresponding shape regularizer uses the symmetric squared-log form

$$\mathcal{L}^{\mathrm{MIMIC}}_{\mathrm{shape}} = \frac{1}{K(N-1)} \sum_{k=1}^{K} \sum_{i=1}^{N-1} \left[\log\left(r_i^{(k)} + 10^{-8} \right) \right]^2, \tag{107}$$

rather than the one-sided log-rate barrier. Accordingly, $\lambda_{\mathrm{rate/shape}}$ in the MIMIC-III row multiplies Equation 107.

Here, $\omega_t \in [0, 1]$ denotes the training warm-up factor used by the USHCN runner. The conditional Gaussian-NLL contribution therefore increases during warm-up and reaches the configured weight 3×10^{-5}; the corresponding tail-MSE weight is zero.

For M4 Daily, $\widetilde{\mathcal{L}}_{\mathrm{sMAPE}}$ denotes the differentiable sMAPE training term normalized by 100, with coefficient 1.0. For Crypto, $\mathcal{L}_{\mathrm{tail}}$ is the squared-error term over the final portion of the prediction horizon.

Table 9: Dataset-specific SHIFT training objectives and temporal-geometry regularization coefficients. w_{reg} denotes an outer multiplier applied to the complete warp regularizer. For MIMIC-III, $\lambda_{\mathrm{rate/shape}}$ multiplies the squared-log shape penalty in Equation 107; for the other temporal configurations it denotes the log-rate penalty. A dash indicates that the corresponding term is inactive in the dedicated configuration.

DATASET	FORECASTING / AUXILIARY OBJECTIVE	w_{reg}	$\lambda_{\mathrm{smooth}}$	$\lambda_{\mathrm{rate/shape}}$	λ_{KL}	$\lambda_{\mathrm{length}}$	λ_{div}
MIMIC-III	$\mathcal{L}_{\mathrm{base}}$	1	5×10^{-4}	1×10^{-5}	1×10^{-5}	1×10^{-3}	1×10^{-5}
USHCN	$\mathcal{L}_{\mathrm{base}}+\omega_t(3\times10^{-5})\mathcal{L}_{\mathrm{NLL}}$	1	2×10^{-3}	1×10^{-5}	0	5×10^{-3}	0
Human Activity	$\mathcal{L}_{\mathrm{base}}$	1	2×10^{-4}	5×10^{-5}	0	1×10^{-4}	4×10^{-3}
MuJoCo	$\mathcal{L}_{\mathrm{MSE}}+0.01\mathcal{L}_{\mathrm{NLL}}+5\mathcal{L}_{\mathrm{floor}}+0.02\mathcal{L}_{\mathrm{calib}}$	0.02	1×10^{-3}	2×10^{-4}	0	1×10^{-3}	3×10^{-3}
M4 Quarterly	$0.70\mathcal{L}_{\mathrm{base}}+0.30\mathcal{L}_{\mathrm{sMAPE}}$	1	2×10^{-3}	1×10^{-5}	0	5×10^{-3}	0
M4 Weekly	$\mathcal{L}_{\mathrm{base}}$	1	2×10^{-3}	1×10^{-5}	0	5×10^{-3}	0
M4 Daily	$\widetilde{\mathcal{L}}_{\mathrm{sMAPE}}+0.10\mathcal{L}_{\mathrm{base}}$	0.10	2×10^{-3}	1×10^{-5}	0	5×10^{-3}	0
Crypto	$0.88\mathcal{L}_{\mathrm{MSE}}+0.10\mathcal{L}_{\mathrm{NLL}}+0.02\mathcal{L}_{\mathrm{tail}}$	1	5×10^{-3}	5×10^{-5}	1×10^{-6}	5×10^{-2}	0
Player Trajectory	$\mathcal{L}_{\mathrm{base}}$	1	2×10^{-3}	1×10^{-5}	0	5×10^{-3}	0
Shallow Water	$5\mathcal{L}_{\rho}+50\mathcal{L}_{u}+50\mathcal{L}_{v}$	0.05	1×10^{-3}	2×10^{-4}	–	–	–

Table 10: Monte Carlo temporal-geometry sampling budgets used in the SHIFT configurations. K_{train} and K_{eval} denote the numbers of sampled temporal geometries used during training and evaluation, respectively. These values are distinct from any sampling budget used only for computational profiling.

DATASET	K_{train}	K_{eval}
MIMIC-III	2	8
USHCN	2	8
Human Activity	4	32
MuJoCo	6	32
M4 Quarterly	1	1
M4 Weekly	2	8
M4 Daily	2	16
Crypto	4	64
Player Trajectory	2	8
Shallow Water	1	8

For MuJoCo, $\mathcal{L}_{\mathrm{floor}}$ discourages collapse of the sampled temporal uncertainty, while $\mathcal{L}_{\mathrm{calib}}$ aligns total predictive uncertainty with detached realized forecasting error.

The MIMIC-III regularization coefficients shown in Table 9 correspond to the configured `PTG_LAMBDA_*` defaults used by the implementation.

Table 10 reports the numbers of stochastic temporal geometries used during training and evaluation.

E.6 Computational Resources and Model Complexity

Model complexity. Table 11 reports the number of trainable parameters for each dataset-specific SHIFT configuration. The parameter count varies across benchmarks because the input and output dimensions and the retained KRNO backbone configuration are adapted to the corresponding forecasting task. All temporal forecasting configurations contain fewer than 0.42 million trainable parameters. The shallow-water configuration is larger because it operates on spatio-temporal physical fields with a higher-capacity operator backbone.

Reproducibility and implementation settings. The following tables summarize the principal dataset-specific optimization and forecasting settings used in the reported experiments. Epoch

Table 11: Trainable parameter counts for the dataset-specific SHIFT configurations.

DATASET	PARAMETERS
MIMIC-III	413,894
USHCN	351,780
Human Activity	179,670
MuJoCo	181,574
M4 Quarterly	172,892
M4 Weekly	172,900
M4 Daily	172,900
Crypto	172,900
Player Trajectory	173,255
Shallow Water	3,961,238

Table 12: Optimization settings for the irregular forecasting experiments. Epochs denote the maximum optimization budget, Batch denotes the physical mini-batch size, and Accum. denotes the number of gradient-accumulation steps.

DATASET	EPOCHS	BATCH	ACCUM.	OPTIMIZER	LR	WD
MIMIC-III	20	1	16	AdamW	3×10^{-4}	10^{-6}
USHCN	70	1	8	AdamW	5×10^{-4}	10^{-6}
Human Activity	30	1	4	AdamW	2×10^{-5}	10^{-6}

counts denote maximum optimization budgets rather than the epoch selected by early stopping. Physical mini-batch size and gradient accumulation are reported separately. Model selection is based on validation performance, after which the selected model is used for evaluation. Monte Carlo temporal-geometry sampling budgets are reported separately in Table 10.

Irregular forecasting. Table 12 summarizes the principal optimization settings for MIMIC-III, USHCN, and Human Activity. For Human Activity, the reported values correspond to the final fine-tuning stage.

Learning-rate scheduling and optimization-control settings for the same three benchmarks are reported in Table 13.

MuJoCo. MuJoCo is evaluated under both regularly observed inputs and controlled levels of randomly removed observations. The same optimization configuration is used across the evaluated observation regimes. Table 14 summarizes the corresponding training and forecasting settings.

Crypto. Crypto uses a regular multivariate forecasting protocol with a fixed input length and prediction horizon. Table 15 summarizes the principal training and forecasting settings.

Player Trajectory. Player Trajectory uses a regular multivariate forecasting protocol with a fixed input length and prediction horizon. Table 16 summarizes the principal optimization and forecasting settings.

M4 forecasting settings. The three evaluated M4 subsets use frequency-specific input and forecast windows together with subset-specific optimization configurations. Table 17 summarizes the principal settings used for Quarterly, Weekly, and Daily.

Shallow Water. Shallow Water uses a spatio-temporal neural-operator configuration on a 32×32 spatial grid with a five-step input context and a five-step prediction window. Unlike the temporal

Table 13: Learning-rate scheduling, early-stopping, and gradient-clipping settings for the irregular forecasting experiments.

DATASET	SCHEDULER	FACTOR	SCHED. PAT.	ES PAT.	CLIP
MIMIC-III	ReduceLROnPlateau	0.8	3	10	1.0
USHCN	ReduceLROnPlateau	0.5	2	12	1.0
Human Activity	ReduceLROnPlateau	0.5	2	8	0.3

Table 14: Training and forecasting configuration used for the MuJoCo experiments. The same configuration is used for the regular setting and for the 30%, 50%, and 70% randomly dropped-observation settings.

SETTING	VALUE
Maximum epochs	100
Physical batch size	1
Gradient accumulation	8
Optimizer	AdamW
Learning rate	7×10^{-4}
Weight decay	10^{-6}
LR scheduler	ReduceLROnPlateau
Scheduler factor	0.5
Scheduler patience	2
Early-stopping patience	10
Input sequence length	50
Forecast horizon	10
Integral channels	64
Observation-drop rates	$0\%, 30\%, 50\%, 70\%$

forecasting benchmarks above, optimization uses Adam with cosine learning-rate annealing. Table 18 summarizes the principal architecture and optimization settings.

The shallow-water model is optimized using the weighted quadrature-based objective

$$\mathcal{L} = 5\mathcal{L}_\rho + 50\mathcal{L}_u + 50\mathcal{L}_v + 0.05\mathcal{L}_{\mathrm{PTG}},$$

where $\mathcal{L}_\rho$, $\mathcal{L}_u$, and $\mathcal{L}_v$ denote the forecasting losses for the three predicted physical fields.

Hardware. All experiments are executed on a single NVIDIA GeForce GTX 1080 Ti GPU with 11 GB of memory.

Computational cost. In addition to the parameter counts reported in Table 11, we characterize the inference-time computational requirements of SHIFT using floating-point operation estimates, end-to-end inference latency, and peak allocated GPU memory. Table 19 reports these measurements for one complete forecast with batch size one.

The reported latency includes the complete uncertainty-aware inference procedure with K_{profile} sampled temporal geometries. Here, K_{profile} denotes the number of samples used for computational profiling and is not necessarily identical to the evaluation setting used for the main forecasting results. In particular, the main shallow-water forecasting experiments use $K_{\mathrm{eval}} = 8$, whereas the computational profile reported in Table 19 was measured using $K_{\mathrm{profile}} = 32$.

The reported values should therefore not be interpreted as the cost of a single deterministic forward pass. Because the datasets differ in sequence length, dimensionality, forecasting horizon, operator configuration, and profiling sample count, the measurements characterize the practical inference cost of each profiled configuration rather than providing a strictly controlled comparison across datasets.

Table 15: Training and forecasting configuration for the Crypto experiments.

SETTING	VALUE
Maximum epochs	10
Physical batch size	1
Gradient accumulation	8
Optimizer	AdamW
Learning rate	4.5×10^{-4}
Weight decay	10^{-6}
LR scheduler	ReduceLROnPlateau
Scheduler factor	0.5
Scheduler patience	2
Early-stopping patience	5
Gradient clipping	1.0
Input sequence length	80
Forecast horizon	15

Table 16: Training and forecasting configuration for the Player Trajectory experiments.

SETTING	VALUE
Maximum epochs	10
Physical batch size	1
Gradient accumulation	8
Optimizer	AdamW
Learning rate	4.5×10^{-4}
Weight decay	10^{-6}
LR scheduler	ReduceLROnPlateau
Scheduler factor	0.5
Scheduler patience	2
Early-stopping patience	5
Gradient clipping	1.0
Input sequence length	30
Forecast horizon	30

Latency and peak allocated GPU memory were measured on an NVIDIA GeForce GTX 1080 Ti. For MIMIC-III and Human Activity, the measurements were obtained from test samples with dataset-dependent sequence lengths; consequently, inference cost may vary across individual examples. GFLOPs were estimated using just-in-time operation tracing. For the shallow-water configuration, the reported GFLOPs are a lower bound because some custom operator and element-wise operations are not supported by the profiling tool. The measurements characterize inference cost only; training time and training-time memory consumption are not included.

Figure 3 visualizes the measured end-to-end inference latency across the profiled SHIFT configurations. The temporal forecasting configurations require between 0.175 and 21.361 GFLOPs per complete forecast, with measured latencies ranging from 12.05 to 416.94 ms. Their peak allocated GPU memory remains below 50 MB. The shallow-water configuration has a substantially higher computational cost because it operates on multivariate physical fields defined over a 32×32 spatial grid and uses a larger spatio-temporal operator backbone.

The reported results also reflect the cost of Monte Carlo temporal-geometry sampling. In general, increasing the number of sampled geometries K requires additional forward computations, although the exact runtime depends on the dataset dimensions, sequence length, implementation, and operator

Table 17: Principal implementation settings for the M4 forecasting experiments.

SETTING	QUARTERLY	WEEKLY	DAILY
Input sequence length	64	104	45
Forecast horizon	8	13	14
Physical batch size	1	1	1
Gradient accumulation	24	16	8
Optimizer	AdamW	AdamW	AdamW
Learning rate	2×10^{-5}	2.5×10^{-4}	6×10^{-4}
Weight decay	10^{-6}	10^{-6}	10^{-6}
LR scheduler	ReduceLROnPlateau	ReduceLROnPlateau	ReduceLROnPlateau
Scheduler factor	0.5	0.5	0.5
Scheduler patience	2	2	2
Gradient clipping	1.0	1.0	1.0

Table 18: Principal training and architecture settings for the Shallow Water experiments.

SETTING	VALUE
Maximum epochs	80
Input sequence length	5
Forecast horizon	5
Physical batch size	4
Optimizer	Adam
Learning rate	2×10^{-4}
Weight decay	5×10^{-7}
LR scheduler	CosineAnnealingLR
Gradient clipping	1.0
Lifting channels	256
Integral channels	48
Projection channels	256
Integral layers	5
Hidden units	96
Hidden layers	5
Spatial grid	32×32

configuration. The profiling measurements characterize the listed K_{profile} configurations but do not establish the minimum number of samples required for stable uncertainty estimation.

F Extended Results

F.1 Per-Dataset Forecasting Results

F.1.1 Irregular Multivariate Forecasting

On USHCN, SHIFT obtains an MSE of $(4.95 \pm 0.03) \times 10^{-1}$, matching the best reported mean achieved by KRNO while reducing the reported standard deviation from 0.08×10^{-1} to 0.03×10^{-1}. SHIFT also achieves the lowest reported USHCN MAE, reducing it from $(3.06 \pm 0.08) \times 10^{-1}$ for KRNO to $(3.00 \pm 0.01) \times 10^{-1}$. Relative to the KRNO rerun mean, this corresponds to an MAE reduction of approximately 2.0%.

On Human Activity, SHIFT reduces the reported MSE from $(2.85 \pm 0.03) \times 10^{-3}$ for KRNO to $(2.77 \pm 0.03) \times 10^{-3}$, and the MAE from $(3.46 \pm 0.02) \times 10^{-2}$ to $(3.29 \pm 0.01) \times 10^{-2}$. These

Table 19: Inference cost per complete forecast with batch size one. K_{profile} is the number of stochastic temporal-geometry samples used for profiling. Latency and peak allocated memory were measured on an NVIDIA GeForce GTX 1080 Ti. The shallow-water GFLOPs are a lower-bound estimate.

DATASET	K_{profile}	GFLOPs	TIME (ms)	MEMORY (MB)
MIMIC-III	32	0.706	142.27	12.8
USHCN	8	1.277	64.29	27.6
Human Activity	32	4.101	149.03	24.3
MuJoCo	32	4.478	181.40	25.6
M4 Quarterly	1	0.175	12.05	29.8
M4 Weekly	8	1.402	63.35	29.8
M4 Daily	8	1.402	61.26	29.8
Crypto	64	21.361	416.94	48.2
Player Trajectory	8	1.001	58.06	23.6
Shallow Water	32	≥ 225.025	1111.01	149.2

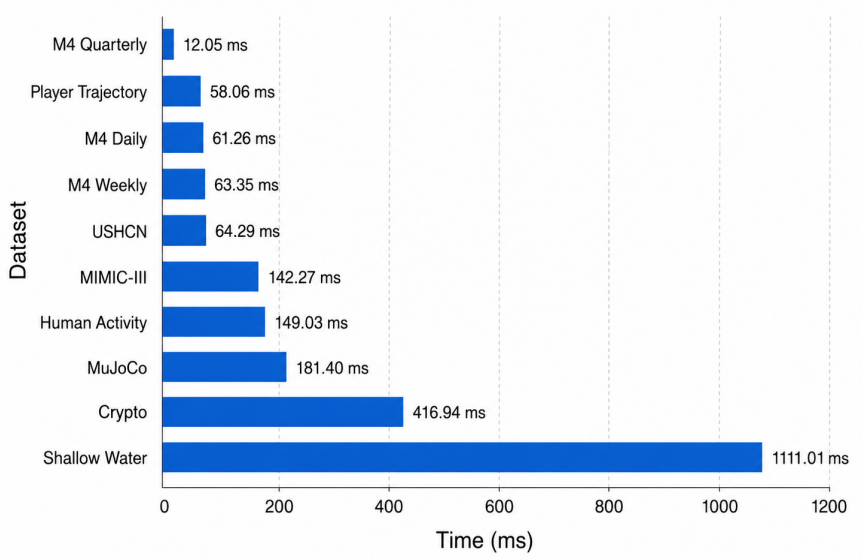


Figure 3: End-to-end inference latency of SHIFT for one complete forecast with batch size one on an NVIDIA GeForce GTX 1080 Ti. The reported latency includes the complete uncertainty-aware inference procedure with the dataset-specific number of sampled temporal geometries K_{profile}. Because the evaluated configurations differ in sequence length, dimensionality, forecasting horizon, operator configuration, and K_{profile}, the measurements characterize the practical inference cost of each configuration rather than a strictly controlled cross-dataset speed comparison.

correspond to relative reductions of approximately 2.8% and 4.9%, respectively. SHIFT therefore ranks second on both Human Activity metrics, behind T-PatchGNN.

The improvement is not uniform across all irregular benchmarks. On MIMIC-III, SHIFT obtains an MSE of $(1.979 \pm 0.03) \times 10^{-2}$ and an MAE of $(7.667 \pm 0.09) \times 10^{-2}$, whereas our KRNO reruns obtain $(1.57 \pm 0.02) \times 10^{-2}$ and $(7.43 \pm 0.06) \times 10^{-2}$, respectively. Thus, the complete SHIFT configuration does not improve point-forecasting accuracy over KRNO on this clinical benchmark under the evaluated protocol.

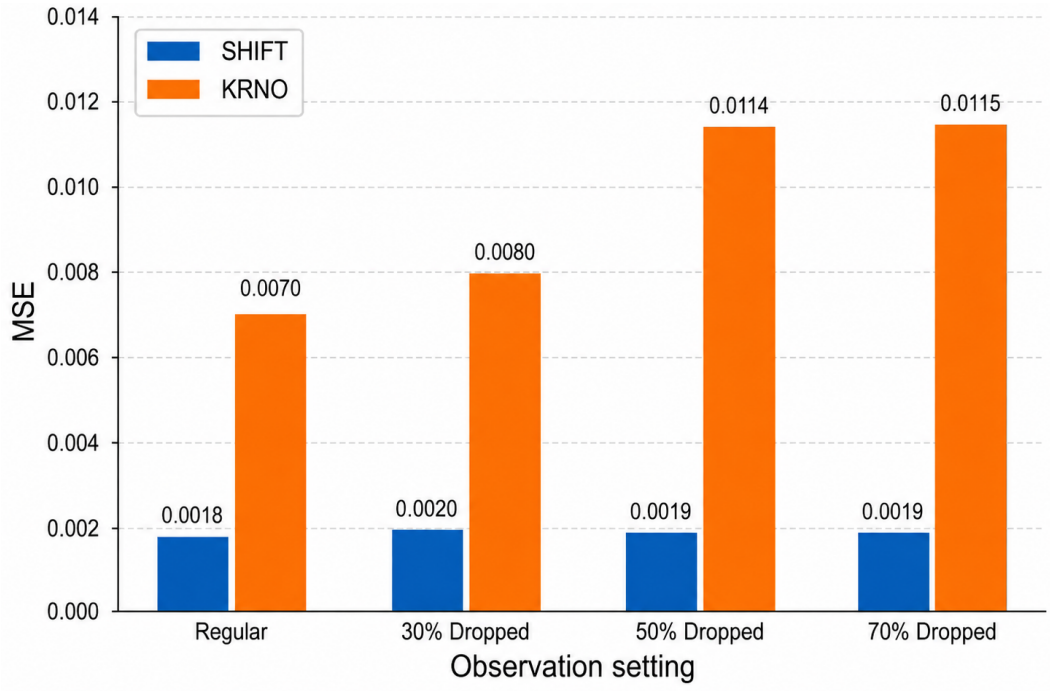


Figure 4: Test MSE of SHIFT and KRNO on MuJoCo under regular sampling and increasing percentages of randomly dropped observations. Lower values indicate better forecasting accuracy. SHIFT achieves substantially lower MSE than KRNO in every evaluated observation regime and remains comparatively stable as the fraction of missing observations increases.

F.1.2 MuJoCo Robustness

Figure 4 visualizes the robustness pattern reported in Table 2. Across all evaluated observation regimes, SHIFT maintains a mean MSE close to 0.002, whereas the corresponding KRNO error increases as the fraction of randomly dropped observations becomes larger.

Detailed MuJoCo comparison. SHIFT achieves the lowest MSE in every observation regime. Under regular sampling, SHIFT reduces the MSE from 0.007 ± 0.002 for KRNO to 0.0018 ± 0.0005, corresponding to a relative reduction of approximately 74.3%. Under 30% missingness, SHIFT obtains an MSE of 0.0020 ± 0.0007, compared with 0.008 ± 0.002 for KRNO, representing a reduction of approximately 75.0%.

Under 50% and 70% missingness, SHIFT maintains MSE values of 0.0019 ± 0.0006 and 0.0019 ± 0.0005, whereas KRNO reaches 0.0114 ± 0.004 and 0.0115 ± 0.002. These correspond to relative reductions of approximately 83.3% and 83.5%, respectively. Within the evaluated random-drop protocol, SHIFT's forecasting error varies only slightly across the tested missingness levels. This result provides evidence of robustness to randomly removed observations under the specific MuJoCo evaluation setting considered here.

F.1.3 Regular Multivariate Forecasting

On Crypto, SHIFT achieves a weighted RMSE of $(2.98 \pm 0.13) \times 10^{-3}$, compared with $(6.91 \pm 0.01) \times 10^{-3}$ for KNF and $(6.95 \pm 0.16) \times 10^{-3}$ for our KRNO rerun. This corresponds to a reduction of approximately 57.1% relative to the KRNO rerun mean.

On Player Trajectory, SHIFT obtains an RMSE of 0.29 ± 0.10, compared with 1.16 ± 0.01 for KNF and 1.25 ± 0.05 for KRNO, corresponding to a relative reduction of approximately 76.8% from the KRNO rerun mean.

Table 20: Identity-warp evaluation on the Shallow Water benchmark. The same trained checkpoint, learned weights, test data, and evaluation procedure are used in both settings.

SETTING	ϵ_ρ	ϵ_u	ϵ_v
Warp ON	0.000418	0.023332	0.022177
Identity warp	0.004169	0.326275	0.337244

These results show that the complete SHIFT architecture can also be effective on regularly sampled financial and trajectory sequences; the effect is therefore not restricted to irregularly sampled observations.

F.1.4 M4 FORECASTING

Table 4 compares SHIFT with the principal M4 baselines. On the Weekly subset, SHIFT achieves the lowest reported sMAPE, with 6.252 compared with the KRNO rerun value of 6.934. This corresponds to a relative reduction of approximately 9.8%.

On Quarterly, SHIFT obtains an sMAPE of 10.408, slightly below the KRNO rerun value of 10.503, corresponding to a reduction of approximately 0.9%. On Daily, SHIFT obtains an sMAPE of 3.041, compared with 3.086 for KRNO. Thus, SHIFT has a lower reported sMAPE than KRNO on all three evaluated M4 subsets, although it does not achieve the strongest overall result on the Quarterly or Daily subsets.

The uncertainty estimates are also reported for all three subsets. Because the Quarterly configuration uses $K_{\mathrm{eval}} = 1$, its between-geometry variance is zero by construction. Accordingly, the reported zero should not be interpreted as evidence that temporal-geometry-induced uncertainty is intrinsically absent on the Quarterly subset; it is not estimable from a single geometry sample. The temporal uncertainty values for Quarterly, Weekly, and Daily are 0, 0.000014, and 0.000015, respectively. The corresponding aleatoric uncertainty values are 0.007690, 0.005658, and 0.004920, while the total uncertainty values are 0.007690, 0.005672, and 0.004936, respectively.

F.1.5 SHALLOW-WATER FORECASTING

Table 5 reports the relative L_2 errors on the shallow-water benchmark. SHIFT achieves the lowest error for the fluid-height field ρ, reducing the KRNO error from 0.001450 to 0.000418. This corresponds to a relative improvement of approximately 71.2%.

For the two velocity fields, KRNO remains the strongest method. SHIFT obtains relative errors of 0.023332 and 0.022177 for u and v, respectively, compared with 0.014970 and 0.014590 for KRNO. Nevertheless, SHIFT ranks second for both velocity variables and remains more accurate than FNO-3D and LOCA. These results show that the effect of stochastic temporal geometry differs across physical state variables: it substantially improves fluid-height forecasting but does not surpass the original KRNO configuration for the velocity components.

Identity-warp evaluation. We evaluate the dependence of the trained predictor on the learned temporal reparameterization by keeping the trained checkpoint, learned weights, test data, and evaluation procedure fixed, and replacing only the learned temporal map by the identity map at inference time.

Replacing the learned temporal map by the identity increases the relative L_2 error from 0.000418 to 0.004169 for ρ, from 0.023332 to 0.326275 for u, and from 0.022177 to 0.337244 for v. Because the trained checkpoint, learned weights, test data, and evaluation procedure are unchanged between the two evaluations, this comparison isolates the effect of replacing the learned temporal reparameterization by the identity map at inference time.

Table 21: Sensitivity of the Crypto uncertainty decomposition to the probabilistic training objective. Values are mean ± standard deviation across repeated runs.

METRIC	CONDITIONAL NLL	FINITE-MIXTURE NLL
U_{temp}	$(8.16 \pm 2.37) \times 10^{-6}$	$(1.06 \pm 0.38) \times 10^{-4}$
U_{alea}	$(4.597 \pm 0.098) \times 10^{-3}$	$(4.661 \pm 0.116) \times 10^{-3}$
U_{total}	$(4.605 \pm 0.100) \times 10^{-3}$	$(4.767 \pm 0.078) \times 10^{-3}$
$U_{\text{temp}}/U_{\text{total}}$	$0.176 \pm 0.048\%$	$2.228 \pm 0.843\%$

F.2 Full Uncertainty Results

Across most datasets, aleatoric uncertainty constitutes the dominant part of the total predictive uncertainty. On Human Activity and USHCN, temporal-geometry-induced uncertainty is on the order of 10^{-7}, whereas the corresponding aleatoric components are on the order of 10^{-3} and 10^{-2}. On MIMIC, the temporal component is negligible at the reported numerical precision.

On MuJoCo, temporal uncertainty increases when observations are removed. It rises from approximately 2×10^{-6} under regular sampling to 7×10^{-6} at 50% missingness, while the aleatoric component remains dominant. Crypto and Player Trajectory show a similar decomposition, with nonzero but comparatively small temporal uncertainty.

The shallow-water experiment shows one of the largest temporal contributions, with temporal uncertainty equal to 1.6920×10^{-4}. Its total uncertainty combines this component with aleatoric uncertainty of 1.292529×10^{-3}. For the M4 Quarterly, Weekly, and Daily subsets, the temporal component remains smaller than the aleatoric component.

F.3 Sensitivity to the Probabilistic Training Objective

To isolate the effect of the probabilistic likelihood formulation, we perform a controlled objective-sensitivity experiment on Crypto. Both configurations retain the same forecasting architecture, sampling budget, optimization settings, and weighted forecasting objective. In particular, the Crypto objective is $0.88\mathcal{L}_{\text{MSE}} + 0.10\mathcal{L}_{\text{prob}} + 0.02\mathcal{L}_{\text{tail}}$. The two configurations differ only in $\mathcal{L}_{\text{prob}}$: the conditional configuration uses $\mathcal{L}_{\text{NLL}}$, whereas the finite-mixture configuration uses $\mathcal{L}_{\text{mix}}$ from Eq. equation 78.

Replacing the component-wise conditional objective with the finite-mixture likelihood increases the mean temporal-geometry uncertainty by approximately $12.9\times$, while the conditional value-uncertainty component remains of similar magnitude. The fraction of represented predictive variance attributed to temporal geometry increases from approximately 0.18% to 2.23%. Thus, the small U_{temp} values observed under conditional training are partly sensitive to the likelihood formulation. At the same time, conditional value uncertainty remains the dominant component on Crypto under both objectives. These results support interpreting U_{temp} as uncertainty represented by the trained model under a specified probabilistic objective, rather than as an objective-independent property of the underlying dataset.

F.4 Probabilistic Calibration

Figure 5 shows the complete empirical calibration curves for Crypto and USHCN over the evaluated nominal coverage levels.

Figure 6 provides a compact visualization of the scalar probabilistic metrics reported in Table 8. Crypto exhibits the smaller mean absolute calibration error, while USHCN achieves particularly close agreement with the nominal target at the 95% coverage level. The NLL magnitudes are interpreted only within each benchmark because they depend on the scale and distribution of the corresponding target variables. These results remain descriptive because matched probabilistic baseline reruns are not available.

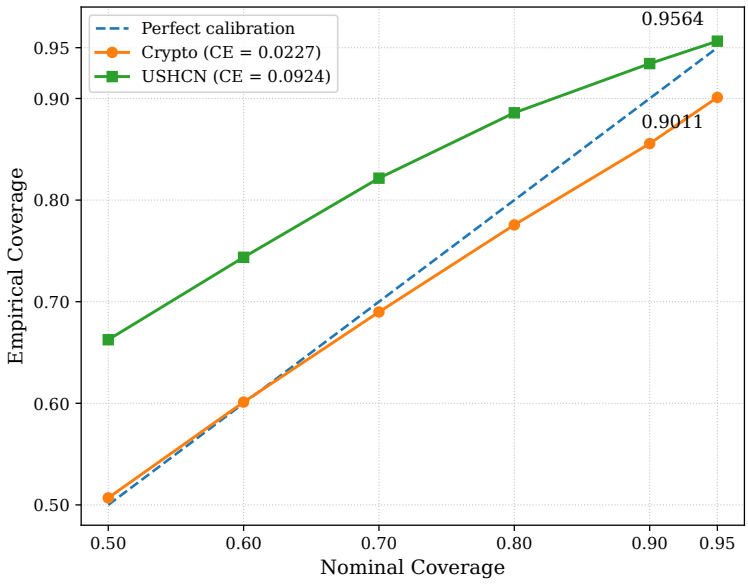


Figure 5: Calibration curves of SHIFT on the regularly sampled Crypto benchmark and the irregularly sampled USHCN benchmark. The dashed diagonal line represents ideal calibration, where empirical coverage equals nominal coverage. Crypto closely follows the ideal line at lower and intermediate confidence levels and exhibits moderate undercoverage at the highest evaluated level. USHCN shows a conservative coverage profile at lower confidence levels and approaches the nominal target as the confidence level increases, achieving empirical coverage of 0.9564 for the nominal 95% interval. CE denotes the mean absolute calibration error over the evaluated nominal levels.

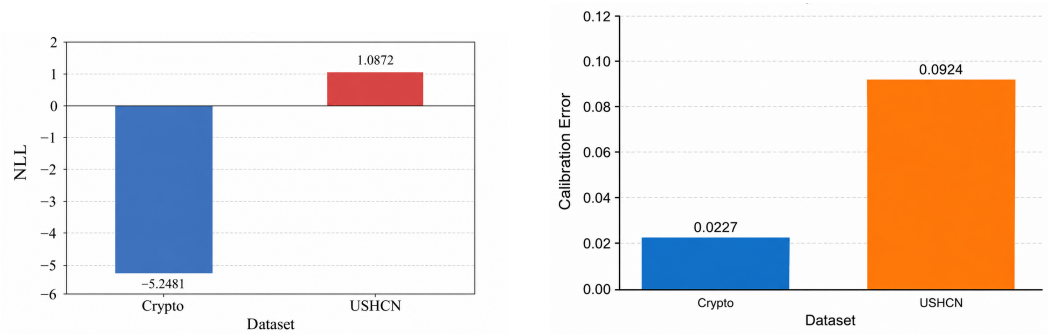


Figure 6: Scalar probabilistic forecasting metrics of SHIFT on Crypto and USHCN. Left: finite-mixture negative log-likelihood (Mix-NLL), where lower values indicate better probabilistic fit within a given benchmark. Because NLL depends on the scale and distribution of the target variables, its absolute values should not be compared directly across datasets. Right: mean absolute calibration error over the evaluated nominal coverage levels $\{0.50, 0.60, 0.70, 0.80, 0.90, 0.95\}$, where lower values indicate closer average agreement between empirical and nominal coverage. The results are descriptive because matched probabilistic baseline reruns are not available for these metrics.

Detailed Crypto and USHCN calibration analysis. On Crypto, SHIFT obtains a finite-mixture NLL of -5.2481, a Gaussian-mixture CRPS of 0.000625, and a 95% mixture-quantile interval score of 0.007325. The empirical coverages at nominal levels of 50%, 60%, and 70% are 0.5069, 0.6012, and 0.6899, respectively. The mean absolute calibration error over the evaluated nominal levels is 0.0227.

At the nominal 95% level, the empirical Crypto coverage is 0.9011, indicating undercoverage of 0.0489. Thus, agreement with the nominal probabilities varies across the evaluated coverage levels: the lower and intermediate levels are closer to their nominal targets, whereas the highest evaluated

level exhibits moderate undercoverage. These results provide a descriptive assessment over the reported nominal levels and should not be interpreted as evidence of uniform calibration outside this evaluated range.

On USHCN, SHIFT obtains a finite-mixture NLL of 1.0872, a Gaussian-mixture CRPS of 0.241554, and a 95% mixture-quantile interval score of 3.602832. Its empirical 95% coverage is 0.9564, corresponding to an absolute difference of 0.0064 from the nominal target.

At the lower and intermediate nominal levels, the empirical USHCN coverage is generally higher than the requested coverage, indicating conservative prediction intervals at those levels. The mean absolute calibration error over all evaluated nominal probabilities is 0.0924. Therefore, the close agreement observed at the 95% level should not be interpreted as evidence of uniform calibration across all evaluated confidence levels.

Overall, calibration is dataset-dependent. Crypto is relatively close to the nominal targets at lower and intermediate coverage levels but exhibits moderate undercoverage at the 95% level. USHCN is conservative at several lower coverage levels while remaining close to the nominal 95% target.

F.5 Full Ablation Results

Ablation setup and variant definitions. We consider five reduced variants: removal of the multi-scale module, removal of adaptive gating, replacement of the stochastic temporal warp with a deterministic warp, replacement of the learned temporal reparameterization with a fixed-coordinate configuration, and removal of smoothness regularization.

The fixed-coordinate variant retains the original physical-time coordinates and the corresponding base measure $d\mu(s) = ds$. Specifically, the sampled temporal map is not used to modify the operator coordinates, latent resampling coordinates, or encoded representation. The forecasting operator therefore receives the original temporal grid together with the unwarped encoded features. This configuration removes learned temporal reparameterization from the forecasting path while retaining the remaining SHIFT components.

The benchmark-specific point-forecasting metrics are retained throughout the ablation analysis. USHCN is evaluated using MSE and MAE, MuJoCo using MSE, Crypto using weighted RMSE, and Player Trajectory using RMSE. Lower values indicate better forecasting accuracy.

Table 22 reports the corresponding uncertainty decomposition for the complete and ablated SHIFT configurations.

Overall forecasting performance. Across all four ablation benchmarks, the complete SHIFT configuration achieves the lowest mean point-forecasting error among the evaluated variants. None of the five reduced configurations improves upon the corresponding complete-model mean, although the magnitude of the degradation is strongly dataset-dependent.

Multi-scale temporal processing. Removing the multi-scale module produces the largest forecasting degradation on both USHCN and Crypto. On USHCN, the no-multi-scale variant obtains an MSE of 5.11 ± 0.05 and an MAE of 3.15 ± 0.04, corresponding to approximately 3.2% and 5.0% increases over the complete-model means, respectively. These are the highest mean USHCN errors among the evaluated ablations.

On Crypto, removing multi-scale processing increases the weighted RMSE from 2.98 ± 0.13 to 3.13 ± 0.10, corresponding to an approximately 5.0% increase in the mean. This is also the largest degradation among the evaluated Crypto ablations.

The contribution of multi-scale processing remains substantial on the dynamical-system and trajectory benchmarks. On MuJoCo, the no-multi-scale configuration obtains an MSE of 0.00307 ± 0.00031, approximately 53.5% above the complete-model mean. On Player Trajectory, removing multi-scale processing increases RMSE from 0.29 ± 0.10 to 0.40 ± 0.03, corresponding to an approximately 37.9% increase in the mean.

These results indicate that representing temporal structure over multiple receptive-field scales contributes to forecasting performance across both regular and irregular observation regimes. Its relative

Table 22: Uncertainty decomposition of the complete and ablated SHIFT configurations on the four ablation benchmarks. Values are reported as mean ± standard deviation across repeated runs. Total uncertainty is computed as $U_{\text{total}} = U_{\text{alea}} + U_{\text{temp}}$. Uncertainty magnitudes characterize the represented predictive distribution and are not independently interpreted as lower-is-better metrics. Totals are computed before rounding.

DATASET	MODEL VARIANT	TEMPORAL UNCERTAINTY	ALEATORIC UNCERTAINTY	TOTAL UNCERTAINTY
USHCN	**SHIFT (Full)**	$(1.50 \pm 0.50) \times 10^{-7}$	$(2.2355 \pm 0.3851) \times 10^{-2}$	$(2.2355 \pm 0.3851) \times 10^{-2}$
USHCN	w/o multi-scale	$(3.20 \pm 0.70) \times 10^{-8}$	$(2.2100 \pm 0.1200) \times 10^{-2}$	$(2.2100 \pm 0.1200) \times 10^{-2}$
USHCN	w/o gating	$(4.10 \pm 0.80) \times 10^{-8}$	$(2.2100 \pm 0.1100) \times 10^{-2}$	$(2.2100 \pm 0.1100) \times 10^{-2}$
USHCN	deterministic warp	0	$(2.2200 \pm 0.1300) \times 10^{-2}$	$(2.2200 \pm 0.1300) \times 10^{-2}$
USHCN	fixed-coordinate	0	$(2.2100 \pm 0.1200) \times 10^{-2}$	$(2.2100 \pm 0.1200) \times 10^{-2}$
USHCN	w/o smoothness	$(3.10 \pm 0.60) \times 10^{-8}$	$(2.2100 \pm 0.1100) \times 10^{-2}$	$(2.2100 \pm 0.1100) \times 10^{-2}$
MuJoCo, 30% Dropped	**SHIFT (Full)**	$(5 \pm 2) \times 10^{-6}$	$(1.363 \pm 0.044) \times 10^{-3}$	$(1.368 \pm 0.046) \times 10^{-3}$
MuJoCo, 30% Dropped	w/o multi-scale	$(1.10 \pm 0.30) \times 10^{-7}$	$(4.9100 \pm 0.1200) \times 10^{-3}$	$(4.9101 \pm 0.1200) \times 10^{-3}$
MuJoCo, 30% Dropped	w/o gating	$(1.00 \pm 0.30) \times 10^{-7}$	$(4.9300 \pm 0.1100) \times 10^{-3}$	$(4.9301 \pm 0.1100) \times 10^{-3}$
MuJoCo, 30% Dropped	deterministic warp	0	$(2.6300 \pm 0.0900) \times 10^{-3}$	$(2.6300 \pm 0.0900) \times 10^{-3}$
MuJoCo, 30% Dropped	fixed-coordinate	0	$(2.0900 \pm 0.0800) \times 10^{-3}$	$(2.0900 \pm 0.0800) \times 10^{-3}$
MuJoCo, 30% Dropped	w/o smoothness	$(1.20 \pm 0.40) \times 10^{-7}$	$(4.9200 \pm 0.1100) \times 10^{-3}$	$(4.9201 \pm 0.1100) \times 10^{-3}$
Crypto	**SHIFT (Full)**	$(2.4 \pm 0.8) \times 10^{-5}$	$(4.839 \pm 0.702) \times 10^{-3}$	$(4.862 \pm 0.761) \times 10^{-3}$
Crypto	w/o multi-scale	$(1.52 \pm 0.25) \times 10^{-5}$	$(4.680 \pm 0.100) \times 10^{-3}$	$(4.695 \pm 0.100) \times 10^{-3}$
Crypto	w/o gating	$(1.46 \pm 0.22) \times 10^{-5}$	$(4.590 \pm 0.090) \times 10^{-3}$	$(4.605 \pm 0.090) \times 10^{-3}$
Crypto	deterministic warp	0	$(4.590 \pm 0.080) \times 10^{-3}$	$(4.590 \pm 0.080) \times 10^{-3}$
Crypto	fixed-coordinate	0	$(4.560 \pm 0.080) \times 10^{-3}$	$(4.560 \pm 0.080) \times 10^{-3}$
Crypto	w/o smoothness	$(8.90 \pm 1.80) \times 10^{-6}$	$(4.570 \pm 0.080) \times 10^{-3}$	$(4.579 \pm 0.080) \times 10^{-3}$
Player Trajectory	**SHIFT (Full)**	$(2 \pm 1) \times 10^{-6}$	$(3.094 \pm 0.005) \times 10^{-3}$	$(3.097 \pm 0.005) \times 10^{-3}$
Player Trajectory	w/o multi-scale	$(1.8 \pm 0.3) \times 10^{-7}$	$(4.408 \pm 0.010) \times 10^{-3}$	$(4.408 \pm 0.010) \times 10^{-3}$
Player Trajectory	w/o gating	$(1.9 \pm 0.3) \times 10^{-7}$	$(4.408 \pm 0.010) \times 10^{-3}$	$(4.408 \pm 0.010) \times 10^{-3}$
Player Trajectory	deterministic warp	0	$(4.408 \pm 0.010) \times 10^{-3}$	$(4.408 \pm 0.010) \times 10^{-3}$
Player Trajectory	fixed-coordinate	0	$(4.408 \pm 0.010) \times 10^{-3}$	$(4.408 \pm 0.010) \times 10^{-3}$
Player Trajectory	w/o smoothness	$(1.9 \pm 0.3) \times 10^{-7}$	$(4.408 \pm 0.010) \times 10^{-3}$	$(4.408 \pm 0.010) \times 10^{-3}$

importance is particularly pronounced for the USHCN climate sequences and Crypto financial sequences.

Adaptive gating. Removing adaptive gating increases the USHCN MSE to 5.07 ± 0.04 and the MAE to 3.12 ± 0.03. On MuJoCo, the same ablation produces an MSE of 0.00304 ± 0.00029, approximately 52.0% above the complete-model mean.

On Crypto, removing adaptive gating increases the weighted RMSE from 2.98 ± 0.13 to 3.05 ± 0.09. On Player Trajectory, the RMSE increases from 0.29 ± 0.10 to 0.38 ± 0.03, corresponding to an approximately 31.0% increase in the mean.

The Player gating ablation produces the smallest mean degradation among the five Player variants, but its forecasting error remains substantially higher than that of the complete configuration. Collectively, the four datasets support the role of adaptive input-dependent scale fusion. The parallel branches provide representations at different temporal receptive-field scales, while the gating mechanism controls their sequence- and geometry-dependent contributions to the final forecast.

Stochastic temporal warping. Replacing the stochastic temporal warp with a deterministic warp removes between-geometry sampling variability and therefore gives $U_{\text{temp}} = 0$ by construction. Despite eliminating this uncertainty component, the deterministic configuration produces higher mean forecasting error on all four datasets.

On USHCN, the deterministic-warp configuration obtains an MSE of 5.08 ± 0.05 and an MAE of 3.14 ± 0.04, compared with 4.95 ± 0.03 and 3.00 ± 0.01, respectively, for the complete model.

The effect is substantially stronger on MuJoCo. The deterministic variant obtains an MSE of 0.00329 ± 0.00033, approximately 64.5% above the complete-model mean.

On Crypto, replacing stochastic warping with a deterministic transformation increases weighted RMSE to 3.03 ± 0.08, approximately 1.7% above the complete-model mean. The effect is larger on

Player Trajectory, where the deterministic variant obtains an RMSE of 0.39 ± 0.03, approximately 34.5% above the complete-model mean.

These findings show that eliminating temporal-geometry variability does not improve point forecasting. The comparatively small magnitude of U_{temp} relative to U_{alea} therefore does not imply that stochastic temporal reparameterization is unimportant. The sampled temporal geometries can affect the predictive mean even when their between-geometry variance constitutes only a small fraction of total predictive uncertainty.

Learned temporal reparameterization. The fixed-coordinate configuration removes learned temporal reparameterization and retains the original physical-time coordinate system throughout the forecasting path. Because alternative stochastic temporal geometries are no longer sampled, this variant also has $U_{\mathrm{temp}} = 0$.

On USHCN, the fixed-coordinate configuration obtains an MSE of 5.06 ± 0.05 and an MAE of 3.13 ± 0.04, both above the complete-model values. On Crypto, its weighted RMSE is 3.00 ± 0.09, compared with 2.98 ± 0.13 for the complete model.

The effect is considerably larger for trajectory dynamics. On MuJoCo, the fixed-coordinate configuration obtains an MSE of 0.00345 ± 0.00036, the largest mean error among the five MuJoCo ablations and approximately 72.5% above the complete-model mean.

On Player Trajectory, the fixed-coordinate configuration obtains an RMSE of 0.40 ± 0.03, approximately 37.9% above the complete-model mean.

These results indicate that adapting the allocation of latent temporal resolution to the observed sequence is particularly important for partially observed dynamical trajectories. The effect is smaller, but remains directionally consistent, on USHCN and Crypto.

Smoothness regularization. Removing smoothness regularization produces a comparatively moderate degradation on USHCN and Crypto. On USHCN, the no-smoothness configuration obtains an MSE of 5.05 ± 0.04 and an MAE of 3.13 ± 0.03. On Crypto, it obtains a weighted RMSE of 3.00 ± 0.08.

The contribution of smoothness regularization is substantially larger on MuJoCo and Player Trajectory. On MuJoCo, removing smoothness regularization produces an MSE of 0.00302 ± 0.00028, approximately 51.0% above the complete-model mean.

On Player Trajectory, removing smoothness regularization produces the largest degradation among all five Player ablations. The RMSE increases from 0.29 ± 0.10 for the complete model to 0.41 ± 0.04, corresponding to an approximately 41.4% increase in the mean.

This pattern is consistent with a stronger contribution of geometric regularity in continuous trajectory and physical-dynamics forecasting. The smoothness penalty discourages abrupt variation between adjacent temporal increments while preserving input-dependent temporal expansion and compression.

Uncertainty decomposition. Across all four datasets, total predictive uncertainty is numerically dominated by the aleatoric component. On USHCN, the temporal component of the stochastic ablations is on the order of 10^{-8}, whereas the corresponding aleatoric component is on the order of 10^{-2}. Consequently, adding the temporal component changes only the final digits of the reported total uncertainty.

The same qualitative pattern appears on MuJoCo. Temporal uncertainty for the stochastic reduced configurations is on the order of 10^{-7}, while aleatoric uncertainty is on the order of 10^{-3}. The deterministic-warp and fixed-coordinate variants have zero temporal uncertainty, while their total uncertainty equals their aleatoric component.

Crypto exhibits a larger relative temporal contribution than USHCN, MuJoCo, and Player Trajectory. The stochastic Crypto ablations have temporal uncertainty between approximately 8.9×10^{-6} and 1.52×10^{-5}, while their aleatoric components remain on the order of 10^{-3}. The aleatoric component therefore remains dominant.

On Player Trajectory, the stochastic ablations have temporal uncertainty on the order of 10^{-7}, compared with aleatoric uncertainty of approximately 4.408×10^{-3}. The complete Player model has total uncertainty of $(3.097 \pm 0.005) \times 10^{-3}$, while the reduced configurations have mean total uncertainties close to 4.408×10^{-3}.

The uncertainty results also illustrate why uncertainty magnitude cannot be interpreted as an independent forecasting criterion. On MuJoCo, the fixed-coordinate configuration has total uncertainty of approximately 2.09×10^{-3}, which is lower than that of several other ablations, yet it produces the largest mean MSE among the reduced MuJoCo configurations.

The same phenomenon appears on Crypto. The fixed-coordinate model has total uncertainty of approximately 4.56×10^{-3}, slightly below the complete-model value reported in Table 6, while its weighted RMSE is higher than that of the complete model.

Thus, a smaller predictive variance does not necessarily correspond to a more accurate predictive mean. Uncertainty magnitude should be interpreted jointly with point-forecasting accuracy and, where available, calibration metrics.

G Failure Cases

SHIFT does not uniformly improve forecasting performance across all evaluated settings. Detailed numerical results are reported in Appendix F; here we focus on the principal failure modes and their possible interpretations.

MIMIC-III. SHIFT does not improve point-forecasting accuracy over the KRNO rerun on MIMIC-III. In addition, the temporal-geometry-induced uncertainty is negligible at the reported numerical precision. One possible explanation is that highly sparse and patient-specific clinical sequences are strongly affected by missingness, measurement noise, and inter-patient variability that cannot be represented adequately by a shared temporal deformation alone. This interpretation remains a hypothesis because the current experiments do not separately identify the effects of missingness, observation noise, and temporal alignment.

Shallow-water velocity fields. The shallow-water results are strongly variable-dependent. Although SHIFT improves forecasting of the fluid-height field ρ, it does not surpass KRNO on the two velocity components. This suggests that a single temporal geometry shared across physical fields may be insufficient when different variables evolve according to different effective temporal scales. Field-specific or jointly coupled spatio-temporal geometries may therefore be more appropriate in this setting.

M4 Quarterly. On M4 Quarterly, SHIFT improves slightly over the KRNO rerun but does not outperform the strongest specialized forecasting baseline. This indicates that stochastic temporal reparameterization does not necessarily replace benchmark-specific inductive biases for every regularly sampled forecasting task.

Probabilistic calibration. The probabilistic experiments also exhibit dataset-dependent calibration behavior. Crypto shows undercoverage at the highest evaluated nominal level, whereas USHCN is conservative at several lower coverage levels. Complete numerical results and calibration curves are reported in Appendix F.4.

Overall, stochastic temporal geometry appears most useful when input-dependent temporal reorganization carries predictive information. Its benefit is weaker when observation noise, missingness, benchmark-specific seasonal structure, or variable-specific physical dynamics dominate the forecasting problem.

H Limitations

First, the learned temporal geometry is not uniquely identifiable from the complete forecasting architecture. Different combinations of the encoder, temporal map, operator kernel, multi-scale branches, gating module, and decoder may produce similar predictions. Moreover, the absolute scale of the

unnormalized interval rates is not identifiable from the normalized temporal map, because a common positive rescaling cancels during normalization. The training objective is not itself scale-invariant, however, because regularization terms acting on pre-normalization variables, rates, and coordinates can select a preferred scale during optimization. The learned map should therefore be interpreted as a predictive temporal representation rather than as recovery of a uniquely observable physical clock.

Second, the current implementation uses a factorized Gaussian distribution over the pre-normalization interval variables. Although normalization couples the resulting interval masses, this parameterization may limit dependencies that can be represented across distant temporal regions.

Third, the same temporal geometry is shared across variables or physical fields. This assumption may be restrictive when different variables evolve according to different effective temporal scales, as suggested by the shallow-water results.

Fourth, the decoder uses conditionally diagonal Gaussian predictive distributions. Heavy-tailed, asymmetric, multimodal, or strongly correlated forecast errors may require richer predictive families.

Fifth, inference cost increases with the number of sampled temporal geometries. The present experiments do not establish the minimum sampling budget required for stable uncertainty estimates across all datasets.

Sixth, temporal-geometry-induced uncertainty is small in several benchmarks, and its absolute magnitude is partly sensitive to the probabilistic training objective. The Crypto objective-sensitivity analysis in Appendix F.3 shows that replacing the component-wise conditional NLL with a finite-mixture likelihood can substantially increase U_{temp}. Accordingly, U_{temp} should be interpreted as the between-geometry predictive variability represented by a particular trained model and objective, rather than as an objective-independent property of the task. Temporal geometry need not be a dominant source of predictive variance in every setting.

Finally, KRNO is rerun directly under the corresponding benchmark protocols, whereas several comparisons with other methods rely on previously published values rather than paired reruns within a single implementation. These comparisons should therefore be interpreted as protocol-matched rather than fully controlled head-to-head experiments.