PROFIT: PROTOTYPE-GUIDED LATENT DIFFUSION FOR FEW-SHOT IRREGULAR TIME SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reliable forecasts are often required for a newly deployed weather station, a newly monitored patient, or a new wearable user before sufficient task-specific history can be collected. This setting is particularly challenging when observations are sparse and asynchronous: an unseen task may differ from the training tasks in both its underlying dynamics and its sampling pattern, whereas existing irregular forecasters typically require substantial within-task history and many few-shot models assume regularly sampled inputs. We study entity-disjoint few-shot forecasting of irregular time series and propose ProFIT, a Prototype-Guided Latent Diffusion framework that infers a continuous-time task representation from a small set of observed events. ProFIT encodes complete meta-training tasks as timestamped variable-value event sets, learns a shared function-latent space, and uses contextual values together with sampling statistics to route an unseen task toward representative latent prototypes. A conditional latent diffusion prior then generates task-specific function latents, while a forecast-alignment objective encourages these latents to preserve predictive semantics. The generated latent is decoded at arbitrary query times to produce probabilistic forecasts. Experiments on USHCN, HumanActivity, and PhysioNet under entity-disjoint K-shot protocols show that ProFIT achieves the lowest point-forecasting errors on USHCN and Activity and remains competitive on PhysioNet. These results provide empirical evidence that prototype-conditioned latent generation is effective for transfer to unseen irregular tasks with limited observations.

INTRODUCTION

Irregular time series arise in weather monitoring, clinical records (Sun et al., 2026), and human activity sensing (Weerakody et al., 2021). Their observation times, intervals, and variable coverage vary across events, so fixed-step models may miss important continuous-time dynamics.

The challenge is greater for few-shot forecasting of an entity-disjoint unseen task, such as a station, subject, or patient. Only a few asynchronous events are available, while the task may differ from training entities in sampling patterns and numerical dynamics. The model must therefore infer latent task dynamics and uncertainty from sparse event-level context rather than merely compensate for a short history.

Recent methods use continuous-time representations, graph dependencies, adaptive aggregation, hierarchical patching, or cross-dataset pretraining (Chen et al., 2023; Zhang et al., 2024; Yalavarthi et al., 2024; Liu et al., 2026; Luo et al., 2025; Gao et al., 2024). They generally assume in-distribution forecasting or longer within-task histories and do not jointly address asynchronous events, entity-disjoint tasks, and extremely small contexts. We evaluate entity-disjoint transfer under the stated K-shot protocols; this is not a separately controlled intervention on sampling-pattern or numerical-dynamics shifts.

We represent complete-task dynamics with a global function latent and use it to guide few-shot adaptation. ProFIT (i) encodes complete tasks as timestamped variable-value events and learns a shared latent space, (ii) combines context values with sampling statistics to route an unseen task

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.

to representative prototypes and condition latent diffusion, and (iii) aligns generated latents with decoded predictive distributions. At test time, only context events and target queries are provided.

The main contributions of this paper are:

- We formulate entity-disjoint few-shot forecasting of irregular time series as conditional generation of a continuous-time task representation from a small set of asynchronous events.
- We propose a sampling-aware prototype-conditioned latent diffusion prior that uses contextual values and sampling statistics for unseen-task latent generation.
- We introduce a forecast-alignment objective that couples latent generation with future predictive distributions to preserve forecasting-relevant semantics.

Under entity-disjoint K-shot protocols on USHCN, Activity, and PhysioNet, ProFIT achieves competitive point-forecasting performance on unseen irregular tasks from limited observations.

Related Work

Irregular Time Series Forecasting

Irregular forecasting must handle nonuniform intervals, variable-specific sampling, and time-varying missingness. Early approaches used time decay, neural differential equations, or continuous-time attention, including GRU-D, Latent ODE, and mTAN (Che et al., 2018; Rubanova et al., 2019; Shukla & Marlin, 2021). Recent methods use continuous-time Transformers, graph dependencies, adaptive aggregation, or hierarchical patching: Contiformer, GraFITi, t-PatchGNN, APN, and Hi-Patch (Chen et al., 2023; Yalavarthi et al., 2024; Zhang et al., 2024; Liu et al., 2026; Luo et al., 2025).

Most of these methods target in-distribution forecasting with sufficient history and familiar test-time sampling. We instead study few-shot transfer to unseen tasks with only a few asynchronous events and possible shifts in sampling and numerical dynamics. ProFIT preserves actual event times, variables, and values while inferring global task dynamics in a cross-task continuous function space.

Few-Shot Time Series Forecasting

Few-shot forecasting exploits cross-task knowledge when an unseen task provides limited context. Classical approaches use rapid adaptation, transferable metrics, or context-conditioned functions, including MAML, Prototypical Networks, and Neural Processes (Finn et al., 2017; Snell et al., 2017; Garnelo et al., 2018; Kim et al., 2019). ST-GFSL uses meta-learning and task adaptation for time series (Lu et al., 2022), while UniTS learns transferable priors through multi-domain pretraining (Gao et al., 2024).

Most existing approaches operate on regular or windowed sequences and do not explicitly model time gaps, event density, or variable coverage. A single deterministic representation may also miss multiple plausible dynamics under extreme sparsity. ProFIT instead learns a global function latent space and uses sampling-aware prototypes to guide latent diffusion from a few asynchronous events.

ProFIT also relates to Neural Processes and diffusion models (Garnelo et al., 2018; Kim et al., 2019; Ho et al., 2020; Song et al., 2021). Unlike direct context conditioning, it builds a shared space from complete tasks, uses prototypes for cross-task routing, and aligns generated latents with future predictive distributions.

Problem Formulation

We study few-shot forecasting of multivariate irregular time series for independent entities such as weather stations, subjects, or patients. Let the meta-training set contain M tasks. Because variables do not share a temporal grid, task m is represented by actual observation events:

$$\mathcal{S}_m = \{(t_i^m, v_i^m, x_i^m)\}_{i=1}^{N_m}, \qquad m = 1, \ldots, M, \tag{1}$$

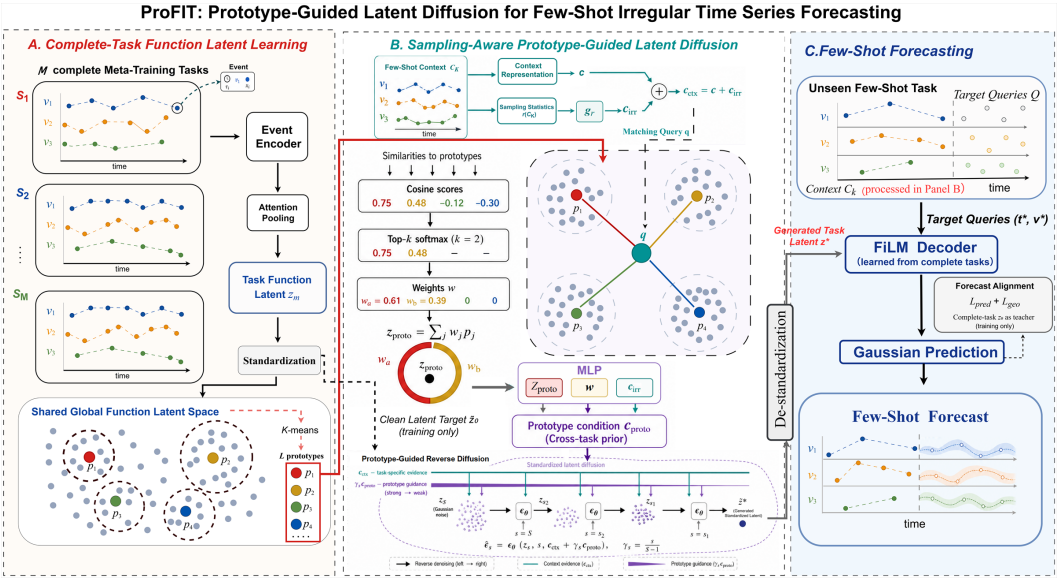


Figure 1: ProFIT framework: complete-task latent learning, sampling-aware prototype guidance, and few-shot probabilistic forecasting.

where $t_i^m \in \mathbb{R}_+$, $v_i^m \in \{1, \ldots, V\}$, $x_i^m \in \mathbb{R}$, and N_m denote the time, variable, value, and event count. Tasks may differ in span, sampling intervals, and variable coverage; no interpolation or shared grid is assumed.

For an entity-disjoint unseen task with forecasting boundary t_b, the model observes only K actual events:

$$\mathcal{C}_K = \{(t_i, v_i, x_i)\}_{i=1}^{K}, \qquad t_i \le t_b. \tag{2}$$

Here, K-shot counts events rather than regular steps or windows. Given post-boundary queries $\mathcal{Q} = \{(t_j, v_j)\}_{j=1}^{N_q}$, the model learns

$$p_\Theta(\mathbf{x}_\mathcal{Q} \mid \mathcal{Q}, \mathcal{C}_K). \tag{3}$$

Queries are real-valued and correspond to actual observations in a fixed future window. The challenge is to infer a task-level function latent z_m when both sampling and numerical dynamics may shift and only K asynchronous events are available. Meta-training latents form a shared function space and transferable prior; at test time, inference uses only $\mathcal{C}_K$ and $\mathcal{Q}$, while $\mathbf{x}_\mathcal{Q}$ is used only for evaluation.

METHOD

ProFIT (Figure 1) learns complete-task function latents, routes a few-shot context through sampling-aware prototypes, generates a task latent by conditional diffusion, and decodes query events probabilistically.

IRREGULAR EVENT ENCODING AND COMPLETE-TASK FUNCTION LATENT LEARNING

ProFIT operates on events without interpolation. For $(t_i^m, v_i^m, x_i^m) \in \mathcal{S}_m$, it uses task-relative time

$$\tau_i^m = t_i^m - t_0^m, \tag{4}$$

where t_0^m is the task origin, removing absolute offsets while preserving relative gaps. Multi-scale logarithmic Fourier features encode the gaps:

$$\gamma(\tau_i^m) = \Big[\sin(2\pi b_k \log(\tau_i^m + \epsilon)), \cos(2\pi b_k \log(\tau_i^m + \epsilon)) \Big]_{k=1}^{F}, \tag{5}$$

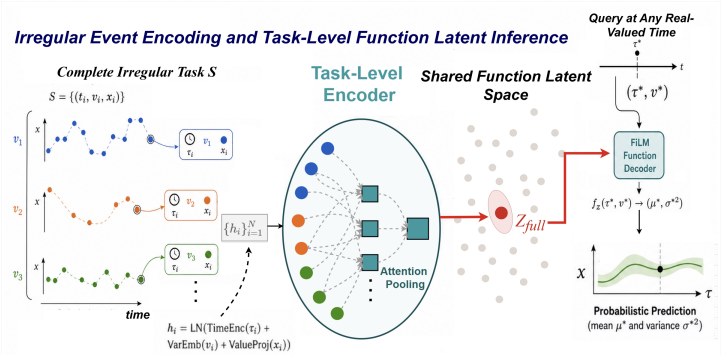


Figure 2: Irregular event encoding and task-level function latent inference.

where b_k and F specify the frequencies and ϵ ensures stability across sampling scales.

The event embedding is

$$h_i^m = \mathrm{LN}(f_t(\gamma(\tau_i^m)) + e_v(v_i^m) + f_x(x_i^m)), \tag{6}$$

where f_t, e_v, and f_x project time, variable, and value. Learnable-query attention pooling gives

$$R_m = \mathrm{vec}\Big(\mathrm{AttnPool}(\{h_i^m\}_{i=1}^{N_m})\Big). \tag{7}$$

Since each event retains γ, pooling is permutation-invariant but time-aware and accommodates varying event counts and variable coverage.

Figure 2 summarizes event encoding and complete-task latent learning.

The complete-task posterior is

$$q_\phi(z_m \mid \mathcal{S}_m) = \mathcal{N}\big(\mu_\phi(\mathcal{S}_m), \mathrm{diag}(\sigma_\phi^2(\mathcal{S}_m))\big). \tag{8}$$

Each $z_m \in \mathbb{R}^{d_z}$ summarizes continuous-time task dynamics. Complete tasks organize the shared latent space and provide teacher representations only during meta-training.

For latent z_m and query (t_j, v_j), a FiLM-modulated decoder outputs

$$p_\theta(x_j \mid t_j, v_j, z_m) = \mathcal{N}(\mu_\theta(t_j, v_j, z_m), \exp(\ell_\theta(t_j, v_j, z_m))). \tag{9}$$

Here, ℓ_θ is the log-variance; FiLM (Perez et al., 2018) conditions the decoder on z_m, and its remaining inputs are the query time and variable. We learn the function space with

$$\begin{aligned}\mathcal{L}_{\text{func}} = \mathbb{E}_{q_\phi(z_m|\mathcal{S}_m)}[-\log p_\theta(\mathbf{x}_{\mathcal{Q}_m} \mid \mathcal{Q}_m, z_m)] \\ + \beta\, \mathrm{KL}[q_\phi(z_m \mid \mathcal{S}_m) \| p(z)],\end{aligned} \tag{10}$$

where $\mathcal{Q}_m$ and $\mathbf{x}_{\mathcal{Q}_m}$ are training queries and values and $p(z) = \mathcal{N}(0, I)$. The likelihood preserves predictive information, the KL regularizes the shared latent space, and free bits mitigate posterior collapse.

The posterior is a meta-training teacher only. Test inference uses $\mathcal{C}_K$, $\mathcal{Q}$, and fixed prototypes and standardization statistics, never $\mathcal{S}_m$ or target values.

Sampling-Aware Prototype-Guided Latent Diffusion Prior

Sparse contexts may admit multiple dynamics, motivating a conditional latent distribution. Let $\bar{z}$ and s_z be elementwise statistics of the meta-training posterior means. ProFIT encodes the context and clusters the standardized means:

$$c = g_\psi(\{h_i : (t_i, v_i, x_i) \in \mathcal{C}_K\}) \in \mathbb{R}^{d_c}, \tag{11}$$

$$\tilde{\mu}_m = (\mu_\phi(\mathcal{S}_m) - \bar{z}) \oslash s_z, \tag{12}$$

$$\mathcal{P} = \mathrm{Cluster}(\{\tilde{\mu}_m\}_{m=1}^M) = \{p_l\}_{l=1}^P, \qquad p_l \in \mathbb{R}^{d_z}. \tag{13}$$

Each prototype represents a region of task dynamics, not a particular task or final prediction.

The matcher augments c with context-only sampling statistics:

$$r(\mathcal{C}_K) = \big[\rho_{\text{valid}}, \tilde{K}, \Delta t, \mu_{\delta t}, \sigma_{\delta t}, f_1, \ldots, f_V\big], \tag{14}$$

where the entries are valid-event ratio, normalized count, observed span, gap mean and standard deviation, and variable frequencies; undefined gaps are zero. An encoder $g_r : \mathbb{R}^{V+5} \to \mathbb{R}^{d_c}$ produces

$$\begin{aligned} c_{\text{irr}} &= g_r(r(\mathcal{C}_K)) \in \mathbb{R}^{d_c}, \\ c_{\text{ctx}} &= c + c_{\text{irr}}, \\ q &= \text{Norm}(W_q c_{\text{ctx}}) \in \mathbb{R}^{d_z}. \end{aligned} \tag{15}$$

Sampling statistics distinguish similar-valued contexts with different spans, gaps, or coverage. The routing query yields cosine logits; let $\mathcal{I}_k$ index the largest k (with $\mathcal{I}_P = \{1, \ldots, P\}$):

$$\begin{aligned} a_l &= \frac{q^\top \text{Norm}(p_l)}{\tau_p}, \qquad l = 1, \ldots, P, \\ \tilde{a}_l &= \begin{cases} a_l, & l \in \mathcal{I}_k, \\ -\infty, & l \notin \mathcal{I}_k, \end{cases} \qquad w_l = \frac{\exp(\tilde{a}_l)}{\sum_{j=1}^{P} \exp(\tilde{a}_j)}. \end{aligned} \tag{16}$$

Thus $w \in \mathbb{R}^P$, $w_l \geq 0$, and $\sum_l w_l = 1$. The prototype anchor and condition are

$$\begin{aligned} z_{\text{proto}} &= \sum_{l=1}^{P} w_l p_l \in \mathbb{R}^{d_z}, \\ c_{\text{proto}} &= \text{MLP}([z_{\text{proto}}, w, c_{\text{irr}}]) \in \mathbb{R}^{d_c}. \end{aligned} \tag{17}$$

This convex anchor interpolates among learned modes, while w and c_{irr} retain routing confidence and sampling information.

For $z_0 \sim q_\phi(z \mid \mathcal{S}_m)$, define $\tilde{z}_0 = (z_0 - \bar{z}) \oslash s_z$, equivalently $z_0 = \bar{z} + s_z \odot \tilde{z}_0$. Under the cosine schedule, $\alpha_s = 1 - \beta_s$ and $\bar{\alpha}_s = \prod_{r=0}^{s} \alpha_r$, giving (Ho et al., 2020)

$$\tilde{z}_s = \sqrt{\bar{\alpha}_s}\tilde{z}_0 + \sqrt{1 - \bar{\alpha}_s}\epsilon, \quad \epsilon \sim \mathcal{N}(0, I). \tag{18}$$

The denoiser is conditioned by

$$\begin{aligned} \hat{\epsilon} &= \epsilon_\omega(\tilde{z}_s, s, c_{\text{ctx}} + \gamma_s c_{\text{proto}}), \\ \gamma_s &= \frac{s}{S-1}, \qquad s \in \{0, \ldots, S-1\}. \end{aligned} \tag{19}$$

with objective

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{s,\epsilon}\left[\|\epsilon - \hat{\epsilon}\|_2^2\right]. \tag{20}$$

Context remains active, while γ_s weakens prototype guidance from the high-noise mode toward task-specific recovery (Figure 3). Diffusion can therefore represent multiple dynamics compatible with sparse context.

Forecast-Aligned Learning and Few-Shot Probabilistic Forecasting

Because latent proximity need not imply similar forecasts, we recover a clean, de-standardized latent:

$$\begin{aligned} \hat{\tilde{z}}_0 &= \frac{\tilde{z}_s - \sqrt{1 - \bar{\alpha}_s}\hat{\epsilon}}{\sqrt{\bar{\alpha}_s}}, \\ \hat{z}_0 &= \bar{z} + s_z \odot \hat{\tilde{z}}_0. \end{aligned} \tag{21}$$

Forecast alignment uses target events:

$$\mathcal{L}_{\text{pred}} = -\log p_\theta(\mathbf{x}_\mathcal{Q} \mid \mathcal{Q}, \hat{z}_0). \tag{22}$$

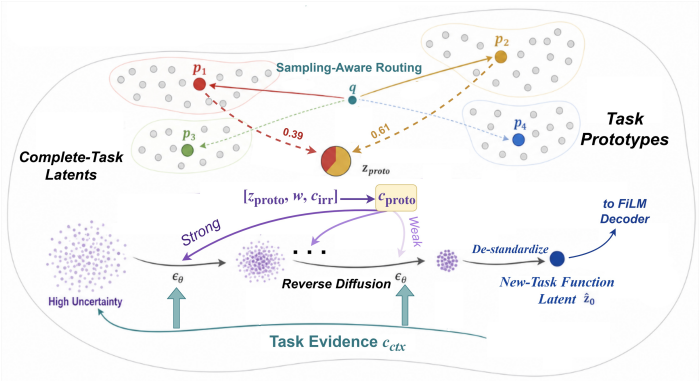


Figure 3: Sampling-aware prototype routing and progressively guided latent diffusion.

and matches teacher and generated predictive distributions:

$$\mathcal{L}_{\text{geo}} = \text{KL}\Big[p_\theta(\cdot \mid \mathcal{Q}, z_0) \,\Big\|\, p_\theta(\cdot \mid \mathcal{Q}, \hat{z}_0)\Big]. \tag{23}$$

This Gaussian KL aligns predictive means and variances; prototype losses are

$$\mathcal{L}_{\text{pa}} = \|z_{\text{proto}} - \tilde{z}_0\|_2^2, \tag{24}$$

$$z_{\text{proto}}^{\text{de}} = \bar{z} + s_z \odot z_{\text{proto}}, \tag{25}$$

$$\mathcal{L}_{\text{pp}} = -\log p_\theta(\mathbf{x}_\mathcal{Q} \mid \mathcal{Q}, z_{\text{proto}}^{\text{de}}). \tag{26}$$

For a mini-batch of B tasks, prototype usage balancing is

$$\bar{w} = \frac{1}{B}\sum_{b=1}^{B} w_b, \qquad \mathcal{L}_{\text{bal}} = \frac{1}{P}\left\|\bar{w} - \frac{1}{P}\mathbf{1}\right\|_2^2. \tag{27}$$

The implemented objective is

$$\begin{aligned}\mathcal{L}_{\text{ProFIT}} = \mathcal{L}_{\text{diff}} &+ \eta_e\big(\lambda_{\text{pred}}\mathcal{L}_{\text{pred}} + \lambda_{\text{geo}}\mathcal{L}_{\text{geo}}\big) \\ &+ \eta_e\big(\lambda_{\text{pa}}\mathcal{L}_{\text{pa}} + \lambda_{\text{pp}}\mathcal{L}_{\text{pp}}\big) + \lambda_{\text{bal}}\mathcal{L}_{\text{bal}}.\end{aligned} \tag{28}$$

Here, η_e linearly warms the forecast and prototype constraints from zero to full strength; balancing remains active.

Stage 1 optimizes the complete-task encoder, posterior, and decoder with $\mathcal{L}_{\text{func}}$, then fixes latent statistics and prototypes. Stage 2 optimizes the context encoder, matcher, and denoiser with $\mathcal{L}_{\text{ProFIT}}$ through the frozen but differentiable decoder. Appendix gives details.

At test time, the predictive distribution is

$$p(\mathbf{x}_\mathcal{Q} \mid \mathcal{Q}, \mathcal{C}_K) = \int p_\theta(\mathbf{x}_\mathcal{Q} \mid \mathcal{Q}, z)\, p_\omega(z \mid \mathcal{C}_K, \mathcal{P})\, \mathrm{d}z. \tag{29}$$

For R DDIM samples $z^{(r)} \sim p_\omega$, the integral becomes the equally weighted Gaussian mixture $R^{-1}\sum_{r=1}^{R} p_\theta(\mathbf{x}_\mathcal{Q} \mid \mathcal{Q}, z^{(r)})$ (Song et al., 2021). Point forecasts average component means, and intervals use the mixture; targets are evaluation-only. Because w is on the simplex, z_{proto} is a bounded convex combination, as proved in the supplement.

Experiments

Experimental Setup

Datasets and protocol. We evaluate ProFIT on three irregular multivariate time-series datasets: USHCN for weather monitoring (Menne et al., 2012), HumanActivity (Activity) for human activity

(a) USHCN

Method	MSE ($\times 10^{-3}$)				RMSE ($\times 10^{-2}$)				MAE ($\times 10^{-2}$)			
	K=1	K=5	K=15	K=30	K=1	K=5	K=15	K=30	K=1	K=5	K=15	K=30
APN	9.203	9.219	8.563	8.340	9.593	9.602	9.254	9.132	6.286	6.095	5.825	5.750
UniTS	7.640	8.252	6.236	8.132	8.741	9.084	7.897	9.018	5.471	5.557	4.789	5.358
ST-GFSL	81.937	86.461	86.566	86.201	28.625	29.404	29.422	29.360	26.156	25.270	25.312	25.313
Hi-Patch	8.640	8.517	7.988	7.773	9.295	9.229	8.938	8.816	5.316	5.326	5.173	5.116
TimeFlow	14.801	14.804	14.604	14.403	12.166	12.167	12.085	12.001	8.589	8.590	8.590	8.590
ASTGI	82.054	17.191	9.014	8.892	28.645	13.111	9.494	9.430	26.244	8.829	5.730	5.587
ProFIT	**5.808**	**5.785**	**5.770**	**5.762**	**7.621**	**7.606**	**7.596**	**7.591**	**4.532**	**4.520**	**4.514**	**4.510**

(b) Activity

Method	MSE ($\times 10^{-2}$)				RMSE ($\times 10^{-1}$)				MAE ($\times 10^{-1}$)			
	K=25	K=50	K=75	K=100	K=25	K=50	K=75	K=100	K=25	K=50	K=75	K=100
APN	1.838	1.843	1.842	1.842	1.356	1.358	1.357	1.357	1.061	1.062	1.062	1.062
UniTS	2.196	2.345	2.348	2.386	1.482	1.531	1.532	1.545	1.118	1.160	1.160	1.170
ST-GFSL	14.144	13.862	13.726	13.504	3.761	3.723	3.705	3.675	3.428	3.387	3.367	3.334
Hi-Patch	1.812	1.792	1.779	1.786	1.346	1.339	1.334	1.336	1.052	1.043	1.040	1.041
TimeFlow	1.900	1.899	1.899	1.899	1.378	1.378	1.378	1.378	1.079	1.078	1.078	1.078
ASTGI	1.690	1.586	1.581	1.588	1.300	1.259	1.257	1.260	0.986	0.982	0.979	0.982
ProFIT	**1.591**	**1.535**	**1.489**	**1.450**	**1.261**	**1.239**	**1.220**	**1.204**	**0.964**	**0.943**	**0.927**	**0.916**

(c) PhysioNet

Method	MSE ($\times 10^{-2}$)				RMSE ($\times 10^{-1}$)				MAE ($\times 10^{-2}$)			
	K=5	K=15	K=30	K=60	K=5	K=15	K=30	K=60	K=5	K=15	K=30	K=60
APN	1.979	1.906	1.823	1.730	1.407	1.381	1.350	1.315	6.864	6.547	6.247	6.909
UniTS	4.639	4.220	3.815	3.277	2.154	2.054	1.953	1.810	17.199	15.748	14.457	12.798
ST-GFSL	6.909	6.909	6.911	6.911	2.628	2.628	2.629	2.629	20.077	20.081	20.081	20.080
Hi-Patch	**0.984**	1.207	**0.912**	0.911	**0.992**	1.099	**0.955**	0.954	**5.599**	**5.308**	**5.008**	**4.646**
TimeFlow	1.113	1.106	0.975	0.942	1.055	1.052	0.987	0.971	5.628	5.384	5.295	5.243
ASTGI	1.969	1.115	0.992	0.957	1.403	1.056	0.996	0.978	9.774	6.852	6.309	6.134
ProFIT	1.145	**1.042**	0.964	**0.891**	1.070	**1.021**	0.982	**0.944**	6.158	5.783	5.518	5.275

Table 1: Unified K-shot forecasting results on USHCN, Activity, and PhysioNet in normalized space. Lower is better.

sensing (Kaluza et al., 2010), and PhysioNet for intensive-care monitoring (Silva et al., 2012). Every observation is represented as an event (t, v, x), and tasks are split by weather station, subject, or patient so that meta-training and meta-test entities are disjoint.

For each unseen task, the model receives K actual asynchronous events and forecasts the subsequent 30% window. Supports are fixed and nested across K; all methods use the same tasks, support events, target events, and training-only normalization. The protocol evaluates entity-disjoint transfer rather than a separately controlled sampling or numerical-dynamics shift.

Implementation. USHCN, Activity, and PhysioNet use $K \in \{1, 5, 15, 30\}$, $\{25, 50, 75, 100\}$, and $\{5, 15, 30, 60\}$, respectively. ProFIT uses 128-dimensional event embeddings, four attention heads, and 100 diffusion steps. Validation tasks select $(P, \eta_{\text{diff}}, d_z) = (4, 1 \times 10^{-3}, 32)$ for USHCN, $(10, 5 \times 10^{-4}, 24)$ for Activity, and $(4, 1 \times 10^{-3}, 32)$ for PhysioNet. The function-latent encoder and diffusion prior are each trained for 60 epochs with AdamW on an NVIDIA GeForce RTX 4050 Laptop GPU (6 GB). We report variable-macro MSE, RMSE, and MAE; RMSE is the square root of the corresponding MSE, and across-K summaries average MSE before deriving RMSE. Probabilistic metrics are reported in the supplementary material.

K-Shot Few-Shot Forecasting Results

We compare ProFIT with APN for irregular forecasting (Liu et al., 2026), TimeFlow as a continuous-time implicit neural representation baseline (Le Naour et al., 2024), ASTGI (Liu et al., 2025), Hi-Patch for irregular multivariate forecasting (Luo et al., 2025), UniTS as a general time-series model (Gao et al., 2024), and ST-GFSL for cross-task few-shot adaptation (Lu et al., 2022). We retain each baseline's original architecture and learning mechanism. Each method receives only the same support events and query times through its native interface; no baseline uses interpolation or forward filling, and all methods share target events and variable-macro metrics.

Table 1 reports all K-shot results in normalized space. Within each dataset, the support sets and target events are fixed across K.

Among all compared baselines, ProFIT achieves the lowest error on USHCN and Activity for every context budget and metric. It also achieves competitive results on PhysioNet. Errors generally decrease as K grows, although the trend varies by dataset. All results are single-run point estimates normalized within each dataset. At the largest context budget, the supplementary 90% prediction intervals achieve coverage of 95.97%, 93.58%, and 92.07% on USHCN, Activity, and PhysioNet, respectively, indicating non-degenerate predictive uncertainty.

Ablation Studies

We remove temporal encoding, prototype guidance, or forecast alignment under the same entity split and nested supports. Table 2 reports Activity averages; complete results are in the supplementary material.

Method	MSE ($\times 10^{-2}$)	RMSE ($\times 10^{-1}$)	MAE ($\times 10^{-2}$)
ProFIT	**1.516**	**1.231**	**9.375**
w/o Proto.	1.616	1.271	9.870
w/o Time	1.878	1.370	10.893
w/o Align.	1.616	1.271	9.875

Table 2: Activity meta-test ablations averaged over four K settings. Lower is better.

The full model is best on all Activity metrics. Removing temporal encoding increases MSE by 23.9%, while removing prototype guidance or forecast alignment increases it by 6.6%, confirming complementary benefits.

Hyperparameter Analysis

We analyze the prototype count P, diffusion learning rate η_{diff}, and latent dimension d_z on PhysioNet validation tasks. Table 3 reports validation averages over K. $P = 4$ and $\eta_{\text{diff}} = 1 \times 10^{-3}$ minimize all metrics; $d_z = 32$ minimizes MSE/RMSE, while $d_z = 48$ gives the lowest MAE, so we retain $d_z = 32$. Other datasets are in the supplementary material.

Factor	Candidate	MSE ($\times 10^{-2}$)	RMSE ($\times 10^{-1}$)	MAE ($\times 10^{-2}$)
P	2	1.220	1.105	6.193
	4	**1.173**	**1.083**	**6.137**
	8	1.199	1.095	6.198
η_{diff}	5×10^{-4}	1.175	1.084	6.166
	1×10^{-3}	**1.173**	**1.083**	**6.137**
	5×10^{-3}	1.176	1.084	6.189
d_z	16	1.234	1.111	6.167
	32	**1.173**	**1.083**	6.137
	48	1.175	1.084	**5.987**

Table 3: PhysioNet validation analysis averaged over four K settings. Bold denotes selected values or factor minima.

Case Study

We study held-out USHCN station 552 under sparse asynchronous observations. The forecast mean follows the target trend, while the 90% interval summarizes predictive uncertainty; target values are used only for evaluation. Detailed numerical results and diagnostic analyses are provided in the appendix.

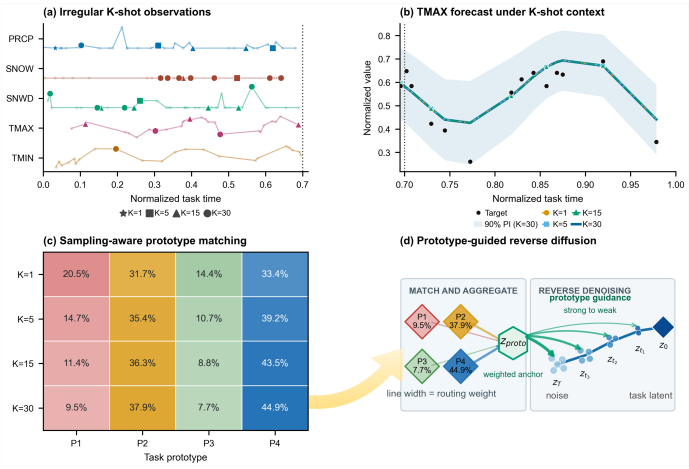


Figure 4: Case study on held-out USHCN station 552: irregular K-shot observations, the $K = 30$ TMAX forecast with a 90% prediction interval, prototype matching, and prototype-guided reverse diffusion.

Efficiency Analysis

We measure Activity inference at $K = 100$, batch size one, and FP32. Table 4 reports parameters, peak memory, latency, MAE, and speedup over DDIM-50 (Song et al., 2021). Latency is the mean and population standard deviation over ten timed repetitions per test task after two task-level warm-ups. DDIM-5 gives an $8.25\times$ speedup with scaled MAE only 0.006 above DDIM-20. APN, TimeFlow, UniTS, and Hi-Patch have lower absolute latency but higher error. Timing-run MAE may differ slightly from Table 1.

Setting	Params	Mem.	Lat. $\mu \pm \sigma$	MAE	Spd.†
ProFIT-5	1.793	11.234	35.891 ± 4.017	9.174	**8.25**$\times$
ProFIT-10	1.793	11.234	58.215 ± 7.225	9.169	5.09$\times$
ProFIT-20	1.793	11.234	122.081 ± 22.704	9.168	2.43$\times$
ProFIT-50	1.793	11.234	296.274 ± 41.010	**9.167**	1.00$\times$
APN	**0.022**	10.902	$\mathbf{1.483 \pm 0.211}$	10.620	–
TimeFlow	0.030	**1.817**	4.021 ± 0.511	10.783	–
UniTS	1.078	21.546	8.762 ± 0.262	11.609	–
ST-GFSL-5	0.202	13.098	1930.319 ± 1233.642	34.433	–
ASTGI	1.517	5140.737	390.353 ± 20.190	10.527	–
Hi-Patch	0.363	32.730	15.107 ± 1.928	10.410	–

Table 4: Activity efficiency at $K = 100$. Parameters, memory, latency, and MAE use millions, MiB, ms/task, and 10^{-2} units; † is speedup over DDIM-50. Bold marks the recommended ProFIT setting or a column best.

Conclusion

We studied few-shot forecasting for unseen irregular entities and proposed ProFIT, which combines event encoding, sampling-aware prototype routing, conditional latent diffusion, and forecast alignment. Under entity-disjoint K-shot protocols, ProFIT is strongest on USHCN and Activity and competitive on PhysioNet, while supplementary intervals provide calibrated uncertainty evidence. These results support prototype-conditioned latent generation for sparse irregular-task transfer, but do not establish robustness to arbitrary distribution shifts.

AI Use Statement

Generative AI tools assisted language editing, formatting, and code debugging; the authors verified all final text, equations, code, and results.

REFERENCES

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.

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

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pp. 1126–1135. PMLR, 2017.

Shanghua Gao, Teddy Koker, Owen Queen, Thomas Hartvigsen, Theodoros Tsiligkaridis, and Marinka Zitnik. UniTS: Building a unified time series model. *arXiv*, 2024. URL `https://arxiv.org/pdf/2403.00131.pdf`.

Marta Garnelo, Dan Rosenbaum, Christopher Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo Rezende, and S. M. Ali Eslami. Conditional neural processes. In *International Conference on Machine Learning*, pp. 1704–1713. PMLR, 2018.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Bostjan Kaluza, Violeta Mirchevska, Erik Dovgan, Mitja Lustrek, and Matjaz Gams. Localization data for person activity. UCI Machine Learning Repository, 2010.

Hyunjik Kim, Andriy Mnih, Jonathan Schwarz, Marta Garnelo, Ali Eslami, Dan Rosenbaum, Oriol Vinyals, and Yee Whye Teh. Attentive neural processes. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=SkE6PjC9KX`.

Etienne Le Naour, Louis Serrano, Léon Migus, Yuan Yin, Ghislain Agoua, Nicolas Baskiotis, Patrick Gallinari, and Vincent Guigue. Time series continuous modeling for imputation and forecasting with implicit neural representations. *Transactions on Machine Learning Research (TMLR)*, 2024.

Xvyuan Liu, Xiangfei Qiu, Huan Cheng, Xingjian Wu, Chenjuan Guo, Bin Yang, and Jilin Hu. ASTGI: Adaptive spatio-temporal graph interactions for irregular multivariate time series forecasting. *arXiv preprint arXiv:2509.23313*, 2025.

Xvyuan Liu, Xiangfei Qiu, Xingjian Wu, Zhengyu Li, Chenjuan Guo, Jilin Hu, and Bin Yang. Rethinking irregular time series forecasting: A simple yet effective baseline. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 23873–23881, 2026.

Bin Lu, Xiaoying Gan, Weinan Zhang, Huaxiu Yao, Luoyi Fu, and Xinbing Wang. Spatio-temporal graph few-shot learning with cross-city knowledge transfer. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1162–1172, 2022.

Y. Luo, B. Zhang, Z. Liu, and Q. Ma. Hi-Patch: Hierarchical patch GNN for irregular multivariate time series. In *Proceedings of the Forty-Second International Conference on Machine Learning*, 2025.

Matthew J. Menne, Imke Durre, Russell S. Vose, Byron E. Gleason, and Tamara G. Houston. An overview of the global historical climatology network-daily database. *Journal of Atmospheric and Oceanic Technology*, 29(7):897–910, 2012.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

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

Satya Narayan Shukla and Benjamin Marlin. Multi-time attention networks for irregularly sampled time series. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=4c0J6lwQ4_.

Ikaro Silva, George Moody, Daniel J. Scott, Leo A. Celi, and Roger G. Mark. Predicting in-hospital mortality of ICU patients: The PhysioNet/computing in cardiology challenge 2012. In *2012 Computing in Cardiology*, pp. 245–248. IEEE, 2012.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in Neural Information Processing Systems*, 30, 2017.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=St1giarCHLP.

Chenxi Sun, Moxian Song, Derun Cai, Baofeng Zhang, Hongyan Li, and Shenda Hong. A review of deep learning methods for irregularly sampled medical time series data. *Health Data Science*, 6:0456, 2026.

Philip B Weerakody, Kok Wai Wong, Guanjin Wang, and Wendell Ela. A review of irregular time series data handling with gated recurrent neural networks. *Neurocomputing*, 441:161–178, 2021.

Vijaya Krishna Yalavarthi, Kiran Madhusudhanan, Randolf Scholz, Nourhan Ahmed, Johannes Burchert, Shayan Jawed, Stefan Born, and Lars Schmidt-Thieme. GraFITi: Graphs for forecasting irregularly sampled time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 16255–16263, 2024.

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

THEORETICAL ANALYSIS

This section formalizes two properties used in the main paper: prototype routing supplies a bounded anchor in the shared function-latent space, and noise prediction learns the conditional latent score. Let $\mathcal{P} = \{p_l\}_{l=1}^{P}$ be the learned prototypes, w the masked-softmax routing weights, and

$$z_{\text{proto}} = \sum_{l=1}^{P} w_l p_l, \qquad d_s = c_{\text{ctx}} + \gamma_s c_{\text{proto}}. \tag{30}$$

Theorem 1 (Bounded prototype anchor and conditional score identity) *The routed anchor belongs to the convex hull of the learned prototypes and satisfies*

$$z_{\text{proto}} \in \text{Conv}(\mathcal{P}), \qquad \|z_{\text{proto}}\|_2 \le \max_l \|p_l\|_2. \tag{31}$$

For the forward process

$$\tilde{z}_s = \sqrt{\bar{\alpha}_s}\tilde{z}_0 + \sqrt{1-\bar{\alpha}_s}\epsilon, \qquad \epsilon \sim \mathcal{N}(0, I), \tag{32}$$

the population minimizer of the conditional noise-prediction loss obeys

$$\begin{aligned} \epsilon^*_\omega(\tilde{z}_s, s, d_s) &= \mathbb{E}[\epsilon \mid \tilde{z}_s, s, d_s], \\ \nabla_{\tilde{z}_s} \log p_s(\tilde{z}_s \mid d_s) &= -\frac{\epsilon^*_\omega(\tilde{z}_s, s, d_s)}{\sqrt{1-\bar{\alpha}_s}}. \end{aligned} \tag{33}$$

Thus, routing provides a bounded coarse-grained location, while the denoiser estimates the score of the task-conditional function-latent distribution.

Proof. Softmax yields $w_l \ge 0$ and $\sum_l w_l = 1$, so z_{proto} is a convex combination of the prototypes. The triangle inequality gives

$$\left\| \sum_l w_l p_l \right\|_2 \le \sum_l w_l \|p_l\|_2 \le \max_l \|p_l\|_2, \tag{34}$$

which proves the prototype bound. For fixed $(\tilde{z}_s, s, d_s)$, conditional expectation uniquely minimizes the population squared error in noise prediction. Applying the denoising-score identity to the Gaussian perturbation kernel gives the conditional score identity (Ho et al., 2020). During reverse diffusion, s decreases and $\gamma_s = s/(S-1)$ decreases with it. Prototype guidance is therefore strongest when the latent is highly corrupted and gradually yields to task-specific context as the sample approaches the clean function latent. □

The theorem does not claim that a prototype is itself the final task latent. Instead, the convex-hull property limits the coarse routing signal to regions supported by meta-training tasks, whereas conditional denoising can still move beyond the anchor to recover unseen-task details.

PROTOTYPE CONSTRUCTION AND OPTIMIZATION DETAILS

Prototypes are constructed only from meta-training tasks. For each task, we use the posterior mean $\mu_\phi(\mathcal{S}_m)$ rather than a stochastic posterior sample, standardize it elementwise with the meta-training statistics $(\bar{z}, s_z)$, and run deterministic Euclidean K-means. Initial centers are selected from evenly spaced meta-training task indices. K-means runs for at most 25 iterations and stops when consecutive centers agree under absolute tolerance 10^{-5} and relative tolerance 10^{-4}. The resulting prototypes and standardization statistics are fixed throughout Stage 2 and at meta-test time.

Stage 1 trains the complete-task encoder, posterior, and decoder for 60 epochs with AdamW. Stage 2 trains the context encoder, sampling-statistics encoder, prototype matcher, and diffusion denoiser for another 60 epochs with AdamW, while retaining a differentiable but parameter-frozen decoder for forecast-alignment losses. Diffusion uses 100 forward steps with a cosine noise schedule, and the auxiliary forecast and prototype losses are introduced by a linear warm-up. Dataset-specific selected values of $(P, \eta_{\text{diff}}, d_z)$ are reported in the main paper and are chosen on validation tasks without meta-test targets.

Variant	MSE	RMSE	MAE
USHCN			
ProFIT	**5.781**	**7.603**	**4.519**
w/o Proto.	6.986	8.358	5.285
w/o Time	8.867	9.416	5.488
w/o Align.	7.072	8.410	5.059
Activity			
ProFIT	**1.516**	**1.231**	**9.375**
w/o Proto.	1.616	1.271	9.870
w/o Time	1.878	1.370	10.893
w/o Align.	1.616	1.271	9.875
PhysioNet			
ProFIT	**1.011**	**1.005**	**5.684**
w/o Proto.	1.070	1.034	5.929
w/o Time	1.046	1.023	5.861
w/o Align.	1.034	1.017	5.758

Table 5: Complete meta-test ablations averaged over four K settings. USHCN reports MSE in 10^{-3} and RMSE/MAE in 10^{-2}; Activity and PhysioNet report MSE/MAE in 10^{-2} and RMSE in 10^{-1}. Lower is better.

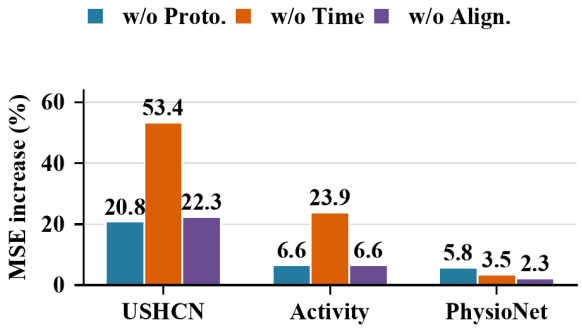


Figure 5: Relative MSE degradation after component removal; positive values indicate worse performance.

Complete Ablation Results

We retrain each ablated variant under the same entity-disjoint split, nested supports, and target events used by the complete model. "w/o Proto." removes prototype matching, prototype conditioning, and prototype auxiliary losses; "w/o Time" removes temporal coordinates from the event and query encoders; "w/o Align." removes the forecast-alignment likelihood and predictive-distribution matching. Table 5 reports meta-test metrics averaged over the four dataset-specific K settings; the later hyperparameter tables report validation metrics instead.

The complete model gives the lowest MSE, RMSE, and MAE on all three datasets in the reported table. Temporal encoding is especially important on USHCN and Activity, where removing it increases average MSE by 53.4% and 23.9%, respectively. On PhysioNet, prototype guidance has the largest MSE effect among the three removals. Forecast alignment consistently improves all three metrics.

Full Hyperparameter Analysis

All hyperparameters are selected on validation tasks without using meta-test targets. We vary one factor at a time: prototype count P, latent diffusion learning rate η_{diff}, and function-latent dimension d_z. Tables 6–8 report averages over each dataset's four validation K settings. Bold in the value column marks the retained configuration, whereas bold metric entries mark within-group minima.

Factor	Value	MSE	RMSE	MAE
P	2	5.900	7.681	4.529
	4	**5.781**	**7.603**	**4.519**
	8	5.962	7.721	4.605
η_{diff}	5×10^{-4}	5.957	7.718	4.600
	10^{-3}	**5.781**	**7.603**	4.519
	5×10^{-3}	5.796	7.613	**4.491**
d_z	16	6.666	8.165	5.089
	32	5.781	7.603	4.519
	48	**5.737**	**7.574**	**4.469**

Table 6: USHCN validation hyperparameter analysis. MSE is scaled by 10^{-3} and RMSE/MAE by 10^{-2}.

Factor	Value	MSE	RMSE	MAE
P	8	1.521	1.233	9.393
	10	**1.517**	**1.232**	**9.375**
	12	1.519	1.232	9.387
η_{diff}	10^{-5}	1.520	1.233	9.386
	5×10^{-4}	**1.517**	**1.232**	**9.375**
	10^{-3}	1.522	1.234	9.395
d_z	20	1.591	1.261	9.779
	24	**1.517**	**1.232**	**9.375**
	28	1.581	1.257	9.687

Table 7: Activity validation hyperparameter analysis. MSE/MAE are scaled by 10^{-2} and RMSE by 10^{-1}.

Selection considers overall validation error and stable behavior as K increases rather than a single isolated metric.

Figure 6 normalizes each validation MSE curve by its selected setting to make the dataset-specific scales comparable. Across datasets, the selected operating points lie in stable local regions rather than at brittle extremes. The prototype count controls the granularity of reusable task modes, the diffusion learning rate controls conditional denoising stability, and d_z trades representation capacity against redundant latent degrees of freedom. On USHCN, $d_z = 48$ reduces validation MSE by only 0.8% relative to $d_z = 32$ while increasing the latent dimension by 50%; we therefore retain the more compact $d_z = 32$ representation.

Factor	Value	MSE	RMSE	MAE
P	2	1.220	1.105	6.193
	4	**1.173**	**1.083**	**6.137**
	8	1.199	1.095	6.198
η_{diff}	5×10^{-4}	1.175	1.084	6.166
	10^{-3}	**1.173**	**1.083**	**6.137**
	5×10^{-3}	1.176	1.084	6.189
d_z	16	1.234	1.111	6.167
	32	**1.173**	**1.083**	6.137
	48	1.175	1.084	**5.987**

Table 8: PhysioNet validation hyperparameter analysis. MSE/MAE are scaled by 10^{-2} and RMSE by 10^{-1}.

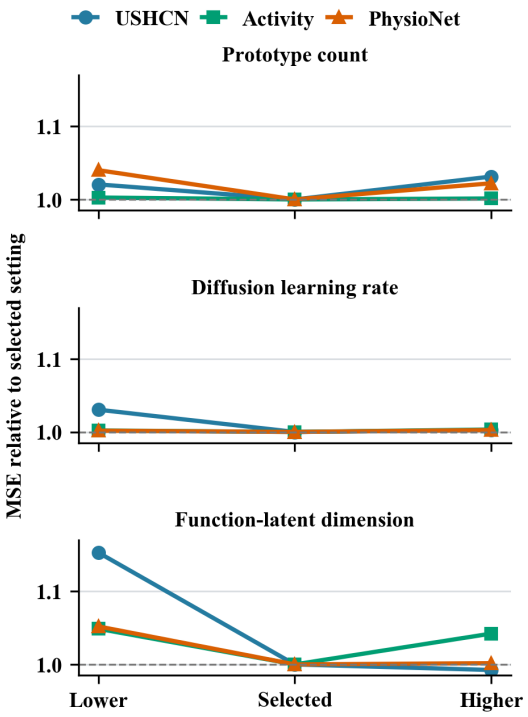


Figure 6: Validation MSE sensitivity normalized by each selected setting. "Lower" and "Higher" are the neighboring candidates in Tables 6–8.

Dataset	K	MSE	RMSE	MAE
USHCN	30	5.762	7.591	4.510
Activity	100	1.450	1.204	0.916
PhysioNet	60	0.891	0.944	5.275

Dataset	NLL	Cov. (%)	Width	Sample Std.
USHCN	-0.890	95.970	0.266	0.0366
Activity	-0.753	93.575	0.412	0.00184
PhysioNet	-1.170	92.070	0.201	0.0233

Table 9: Point and probabilistic metrics at each dataset's largest context budget. Point-metric units follow the main-result table; coverage is for nominal 90% intervals.

Probabilistic Forecasting Results

The decoder outputs a Gaussian predictive distribution at every queried time-variable pair. Besides point errors, we report Gaussian negative log-likelihood (NLL), coverage and width of the nominal 90% prediction interval, and the standard deviation across latent samples. Table 9 uses the largest context budget for each dataset.

Coverage remains close to or moderately above the nominal level on all three datasets. USHCN is conservative, while Activity and PhysioNet obtain narrower dataset-specific intervals with coverage above 90%. These results complement the point metrics by showing that latent diffusion retains meaningful predictive uncertainty rather than producing only a deterministic task representation.

K	MSE	RMSE	MAE
1	2.734	5.229	3.232
5	2.730	5.225	3.228
15	2.727	5.222	3.224
30	**2.719**	**5.214**	**3.222**

Table 10: USHCN station 552 forecasts. MSE is scaled by 10^{-3} and RMSE/MAE by 10^{-2}.

CASE STUDY DIAGNOSTICS

Table 10 provides the point errors corresponding to the held-out USHCN station 552 case study in the main paper. All three errors decrease gradually as the context budget increases, with the lowest values obtained at $K = 30$.