# REPCAL: CALIBRATING TIME-SERIES MODELS WITH FROZEN MULTIMODAL REPRESENTATIONS FOR ZERO-SHOT RECONSTRUCTION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Zero-shot reconstruction aims to reconstruct a target series at locations with no observations, where the inputs are the only source of information about the target. Time-series methods make fuller use of the input series, learning input-to-target mappings that generalize from observed to unobserved locations. Multimodal methods capture more information from modalities beyond time series, such as gridded fields, maps, and imagery, which pretrained encoders turn into representations at every location. Concatenating multimodal representations with the inputs of a time-series model is a straightforward way to combine the two. With limited observations to learn from, such direct fusion improves performance in some scenarios and degrades it in others. Yet for a given model, task, and representation, the errors of direct fusion follow recognizable patterns across locations. This motivates using the representations a second time: as a condition for calibrating the fused model's errors. We propose RepCal, which takes the fused model as a prior and calibrates it with a diffusion model conditioned on the representations, trained on observed locations and transferred to unobserved ones. Experiments on four environmental targets and several families of time-series models demonstrate improvements over direct fusion. Our code is available at https://anonymous.4open.science/r/RepCal-8517/.

## 1 INTRODUCTION

Many environmental variables are measured at only a few locations. Field measurements are expensive, so variables such as streamflow, stream temperature, and snowpack are observed by sparse networks, while many of their drivers, including weather, terrain, and soil, are available much more broadly. Estimating these variables where they have never been measured is therefore a long-standing problem in hydrology (Hrachowitz et al., 2013). We study *zero-shot reconstruction*: estimating the full time series of a target variable at a station where that variable has never been observed, using only information available at that station. Unlike forecasting (Ye et al., 2025), there is no observed history of the target series to extrapolate from. Unlike imputation (Tashiro et al., 2021), there is no observed target value at the station to fill from. When target observations are this sparse, both effective spatial generalization and additional transferable information become important.

One line of work focuses on predicting at unmeasured locations using the observations and covariates that are already available. Recurrent models trained on measured stations can be applied directly to new stations (Kratzert et al., 2019a; Cheng et al., 2026). When measurements from nearby stations are available, neural processes map observations to new locations (Garnelo et al., 2018; Gordon et al., 2020; Nguyen & Grover, 2022; Ashman et al., 2024; 2025), graph models propagate information to unmeasured nodes (Wu et al., 2021; Zheng et al., 2023; Hu et al., 2023; Xu et al., 2025), and diffusion and flow-matching models reconstruct missing locations in space and time (Tashiro et al., 2021; Liu et al., 2023; Islam et al., 2025; Kerrigan et al., 2024; Zhang et al., 2026b; Lalanne et al., 2026; Islam et al., 2026). These methods improve spatial generalization. Their performance, however, is constrained by the information available at observed stations and by how well the learned relationships transfer to the target location.



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.

A complementary direction brings in richer information available beyond the monitoring network. Large-scale observation products provide spatially distributed weather (Thornton et al., 2021), elevation (Gesch et al., 2002), soil properties (Poggio et al., 2021), and other environmental information, while large-sample hydrology increasingly preserves this spatial structure within individual basins (Knoben et al., 2025; Talha et al., 2026). Pretrained encoders can compress such heterogeneous data into reusable representations (Cong et al., 2022; Reed et al., 2023; Tseng et al., 2025), including representations available over broad geographic domains (Klemmer et al., 2025; Brown et al., 2025). More broadly, time-series models have begun to incorporate encoded information from other modalities through pretrained language models (Zhou et al., 2023; Jin et al., 2024; Liu et al., 2024b; 2025), text-derived context (Liu et al., 2024a; Wang et al., 2024a; Williams et al., 2025), and multimodal spatio-temporal representations (Li et al., 2024). In hydrology, pretrained Earth-observation representations have been incorporated into recurrent models either in place of conventional static attributes (Qu et al., 2026) or together with them (Tayal et al., 2024).

The challenge is that a pretrained representation may contain useful information without being directly aligned with a particular downstream target. Existing approaches typically rely on the downstream predictor itself to learn how to interpret and fuse this representation from the available task labels. In the zero-shot setting, this supervision is limited to the measured stations, yet the learned fusion must generalize to locations where the target has never been observed. This is particularly difficult when a relatively small set of physically defined inputs is augmented with a high-dimensional latent representation learned for a different pretraining objective. A simple strategy is to align the representation with the time series and concatenate it to the predictor's inputs, but such direct fusion does not consistently improve prediction: in our experiments, it helps some models and tasks while degrading others. This motivates separating the use of the representation for prediction from a second stage that explicitly refines the resulting prediction using the same multimodal context.

We realize this separation with RepCal, a two-stage framework in which a calibrator, learned on the measured stations, connects a pretrained multimodal representation to an ordinary time-series model. We first pretrain a multimodal encoder on four complementary forms of basin information: spatially distributed weather fields, spatially distributed static basin properties, basin-averaged meteorological time series, and tabular basin attributes (Talha et al., 2026; Tran et al., 2025), using self-supervised masked reconstruction (He et al., 2022). Following a query-based interface (Zhang et al., 2026a), the frozen encoder is queried at each station and time step to produce an aligned representation. We concatenate this representation with the conventional inputs of a time-series model—basin-averaged meteorological forcing and selected static basin attributes in our experiments. The time-series model then produces a first estimate of the target series, which we treat as the *prior*.

Our premise is that part of the prediction error is systematic and predictable from the multimodal context. A second stage can therefore learn this structured error where observations are available and transfer the calibration to stations where the target has never been observed. A conditional diffusion model learns the distribution of the target conditioned on the conventional inputs, the aligned multimodal representation, and the prior (Han et al., 2022; Ye et al., 2025).

RepCal combines the two directions above: additional information comes from the multimodal representation, while calibration is learned from measured stations and transferred to unmeasured ones. The multimodal encoder remains frozen, and the calibrator operates on the conventional inputs, aligned representation, and model prediction without requiring changes to the internal architecture of the upstream predictor. It can therefore be applied to a broad class of time-series models that can consume the aligned representation and produce an initial prediction.

Our contributions are as follows.

- We introduce a queryable multimodal representation that aligns heterogeneous spatial information with downstream time-series inputs, providing one frozen representation across several environmental reconstruction tasks.
- We propose RepCal, a prior-centered diffusion calibrator that learns from measured stations to refine predictions conditioned on the multimodal representation, and transfers this calibration to unmeasured stations while providing predictive uncertainty.
- Across four environmental tasks and multiple families of time-series models, we show that directly adding a pretrained multimodal representation does not consistently improve zero-shot reconstruction, while calibration improves the use of multimodal representations.

## 2 RELATED WORK

**Time-series models at unmeasured stations.** Hydrology has long studied prediction at locations without target measurements (Hrachowitz et al., 2013). A widely used approach trains a single recurrent model across many gauged basins and applies it to ungauged ones (Kratzert et al., 2019a). Static basin attributes provide location-specific information, and in the entity-aware LSTM they modulate the input gate (Kratzert et al., 2019b). Out of sample, however, much of the predictive skill can come from meteorological forcing rather than static attributes (Heudorfer et al., 2025).

A second family transfers information from observations at other locations. Neural processes map observed points to predictions at new ones (Garnelo et al., 2018; Gordon et al., 2020; Nguyen & Grover, 2022; Ashman et al., 2024), with Gridded TNP additionally incorporating gridded inputs (Ashman et al., 2025). Graph models propagate information to unmeasured nodes (Wu et al., 2021; Zheng et al., 2023), combine graph structure with neural processes (Hu et al., 2023), or introduce virtual nodes during training to better match the inference graph (Xu et al., 2025). Diffusion and flow-based methods reconstruct unmeasured locations as missing entries of a space–time field (Tashiro et al., 2021; Liu et al., 2023; Islam et al., 2025; Zhang et al., 2026b). DynaSTI extends diffusion-based interpolation to unseen locations (Islam et al., 2026); functional flow matching models distributions over functions (Kerrigan et al., 2024); and RP Flow learns a field that can be queried continuously in space (Lalanne et al., 2026).

Nearby-station methods additionally use measurements from other stations at test time, whereas our zero-shot setting requires reconstruction from information available at the target station itself. We evaluate these methods with the inputs for which they were designed and also use several of them as upstream predictors for our calibrator.

**Additional information in time-series models.** A complementary line of work enriches the predictor with additional covariates or modalities. DeepAR adds time-varying covariates to autoregressive inputs (Salinas et al., 2020); the Temporal Fusion Transformer uses static covariates as context for variable selection and temporal processing (Lim et al., 2021); and TimeXer represents exogenous series as tokens accessed through cross-attention (Wang et al., 2024b). In hydrology, recurrent models can combine multiple meteorological forcing products (Kratzert et al., 2021).

Recent multimodal time-series models increasingly incorporate text (Jiang et al., 2025). Some use pretrained language models as part of the forecasting architecture, including GPT4TS (Zhou et al., 2023), Time-LLM (Jin et al., 2024), UniTime (Liu et al., 2024b), and TimeCMA (Liu et al., 2025); UrbanGPT extends related ideas to spatio-temporal prediction (Li et al., 2024). Others retain a time-series backbone while adding external context such as news (Liu et al., 2024a; Wang et al., 2024a) or textual descriptions of the setting (Williams et al., 2025).

Across these approaches, additional information is typically integrated within the predictor, which must learn from downstream supervision how to use it for the target task. Our method retains this direct fusion but adds a separate calibration stage conditioned on the same representation.

**Pretrained Earth-observation representations.** Earth-observation encoders are pretrained using masked reconstruction (Cong et al., 2022; Reed et al., 2023), masked modelling with contrastive objectives (Tseng et al., 2025), or contrastive learning (Klemmer et al., 2025). They are commonly evaluated using linear probes, nearest neighbors, or fine-tuning on tasks such as classification and segmentation (Tseng et al., 2025). Other systems provide precomputed representations that describe a location, or a location over time, and map them to downstream targets with relatively simple models (Rolf et al., 2021; Brown et al., 2025). In hydrology, pretrained Earth-observation representations have also been used in place of conventional static attributes in recurrent models (Qu et al., 2026).

Our encoder follows the masked-reconstruction paradigm but uses a query-based interface inspired by D4RT (Zhang et al., 2026a). A lightweight decoder answers independent queries by cross-attending to a latent set (Jaegle et al., 2022; Sajjadi et al., 2022), allowing the frozen representation to be read at a specified station, time, and spatial support. We additionally encode two-dimensional cell size, extending the scale-aware idea used in remote-sensing encoders (Reed et al., 2023), because the physical size and aspect ratio of basin pixels vary substantially. This produces a representation aligned with the temporal inputs of downstream time-series models while keeping the pretrained encoder frozen.

**Diffusion as calibration.** Prior-centered diffusion starts from a regression estimate rather than from pure noise and learns to refine that estimate probabilistically (Han et al., 2022; Ye et al., 2025) (Section 3.3); stochastic interpolants provide a related flow-based formulation (Albergo et al., 2025). We use this idea specifically for multimodal zero-shot reconstruction: the initial estimate comes from a time-series predictor that receives a frozen multimodal representation, and the diffusion model serves as a separate representation-conditioned calibrator learned on measured stations and applied to unmeasured ones.

# 3 PRELIMINARIES

## 3.1 ZERO-SHOT RECONSTRUCTION

Let $\mathcal{S}$ be a set of stations, split into observed stations $\mathcal{O}$ and unobserved stations $\mathcal{U}$, with $\mathcal{O} \cap \mathcal{U} = \emptyset$. We work with one year of $T = 365$ days at a time. Every station $s$ has daily exogenous inputs $x_s \in \mathbb{R}^{T \times d_x}$: basin-averaged weather and a short list of static basin attributes, repeated on every day. Every station also has a target series $y_s \in \mathbb{R}^T$, such as streamflow or snow depth. For $s \in \mathcal{O}$ the target is observed on most days. For $s \in \mathcal{U}$ it is never observed. The task is to learn from $\{(x_s, y_s)\}_{s \in \mathcal{O}}$ a model that reconstructs $y_s$ for every $s \in \mathcal{U}$ from inputs that exist at $s$. How we split stations into $\mathcal{O}$ and $\mathcal{U}$ is described in Section 5.1.

## 3.2 SELF-SUPERVISED MULTIMODAL REPRESENTATIONS

In addition to the daily exogenous input $x_s$, a basin $b$ is characterized by multimodal observations $\mathcal{M}_b = \{m_b^{(j)}\}_{j=1}^J$, functions $m_b^{(j)} : \Omega_j \to \mathbb{R}^{d_j}$ each on its own domain $\Omega_j$. In our data, the weather maps are a video on days $\times$ pixels, the static maps of terrain and soil are an image on pixels, the weather series is a multivariate time series on days, and the basin attributes are a table row (Figure 1). Pixel sizes also differ by orders of magnitude between basins (Appendix D). None of these domains matches the index of the target, which is one station and one day.

Self-supervised pretraining learns a representation of such data without labels. A masked autoencoder hides most of the input, and an encoder that reads only the visible part is trained to reconstruct the rest (He et al., 2022). This has become a standard way to pretrain encoders for images, videos and remote-sensing data (Wang et al., 2023; Cong et al., 2022; Reed et al., 2023). D4RT separates the encoder from the way its output is read (Zhang et al., 2026a). An encoder turns a video $V$ into a set of latent tokens, $F = E(V)$, and a light decoder answers each query $q$ by cross-attention into $F$, $P = D(q, F)$. One encoding thus serves queries at any point and time.

A time-series model reads one vector per station and time step. We call a representation aligned when it comes in this form, a series $r_s \in \mathbb{R}^{T \times d_r}$ that can be concatenated to $x_s$. Section 4.1 obtains it by querying a pretrained encoder at each station and day. The problem we study is to learn, on $\mathcal{O}$, a conditional distribution $p(y_s \mid x_s, r_s)$ that still holds on $\mathcal{U}$, while the encoder stays frozen.

## 3.3 DIFFUSION CENTERED ON A PRIOR

A denoising diffusion model (Ho et al., 2020) learns a distribution by adding Gaussian noise to a sample $y^0$ over $K$ steps, $q(y^k \mid y^0) = \mathcal{N}\big(\sqrt{\bar{\alpha}_k}\, y^0,\, (1 - \bar{\alpha}_k) I\big)$, and training a network to undo it. The chain ends near $\mathcal{N}(0, I)$, so sampling starts from pure noise, which wastes a good estimate of the target when one is available. CARD instead replaces the end point with a regression estimate $\hat{y}$ (Han et al., 2022), $q(y^k \mid y^0, \hat{y}) = \mathcal{N}\big(\sqrt{\bar{\alpha}_k}\, y^0 + (1 - \sqrt{\bar{\alpha}_k})\, \hat{y},\ (1 - \bar{\alpha}_k) I\big)$, so the forward process moves the target toward $\hat{y}$ and the reverse process starts near $\hat{y}$ and refines it: the diffusion model calibrates an estimate instead of generating from scratch. ZeroDiff brings this to zero-shot reconstruction (Fan et al., 2026). It builds $\hat{y}$ from the inputs of each station, so it exists at stations without observations, learns the calibration on $\mathcal{O}$, and its denoiser sees the whole series of conditions at once, so each day draws on both earlier and later days.

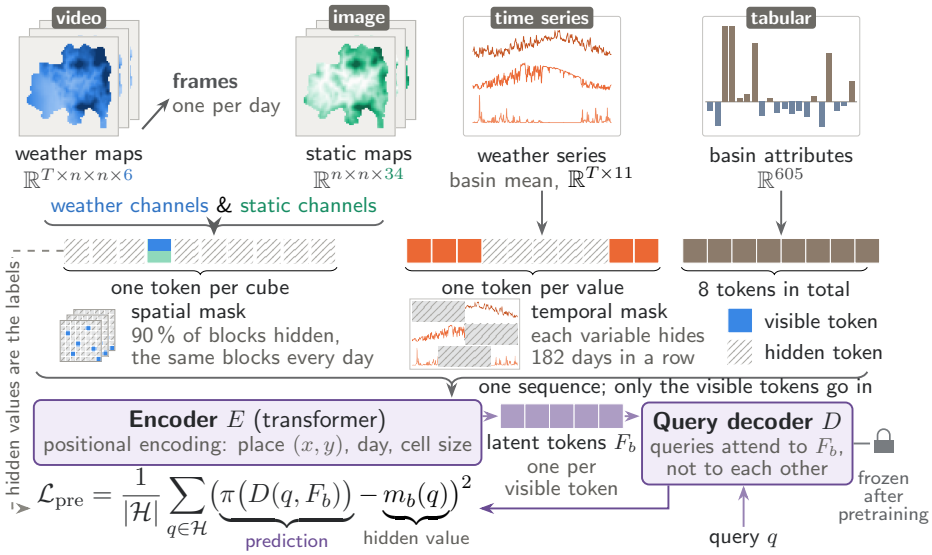


Figure 1: Self-supervised pretraining (Section 4.1). The four inputs of a basin become one sequence of tokens, and most tokens are hidden. The encoder $E$ reads only the visible tokens. The query decoder $D$ answers queries at the hidden values, which serve as labels. Both are then frozen.

# 4 METHOD

RepCal connects a pretrained multimodal representation to an ordinary time-series model in two steps (Figure 2). A frozen encoder, queried at each station and day, gives a representation in the form of the model's inputs. Concatenated to them, it yields a first estimate of the target, the prior. A conditional diffusion model then learns on the observed stations how the prior goes wrong when the representation is present, rather than requiring the prior to make good use of the representation by itself. Figure 1 shows the pretraining of the encoder.

## 4.1 A MULTIMODAL ENCODER THAT CAN BE QUERIED

The encoder $E$ turns the four inputs of a basin (video, image, time series, tabular) into one sequence of tokens, without resampling them to a common grid (Figure 1): one token per space–time block of the weather maps, with the static maps stacked onto every frame, one token per value of the weather series (Nguyen et al., 2023), and a few tokens for the attributes. Each token carries its place, day and cell size, because pixel size differs by orders of magnitude between basins. $E$, a transformer (Dosovitskiy et al., 2021), returns latent tokens $F_b = E(\mathcal{M}_b) \in \mathbb{R}^{N\times d_r}$. The decoder $D$ is a small cross-attention transformer that answers queries, as in D4RT (Zhang et al., 2026a). A query $q = (p, \Delta, t, v)$ asks for variable $v$ on day $t$ in a cell of size $\Delta$ centered at place $p$ (Chen et al., 2021), where $v$ is one of the variables $\mathcal{V}$ of the weather maps and the weather series. It returns a feature $D(q, F_b) \in \mathbb{R}^{d_r}$ by attending to $F_b$ alone.

$E$ and $D$ are pretrained on 9,008 basins by masked reconstruction (He et al., 2022). Most blocks of the maps and half a year of each weather series are hidden, $E$ reads only the visible tokens, and $D$ is queried at a set $\mathcal{H}$ of hidden values $m_b(q)$, which a linear projection $\pi$ predicts with the loss $\mathcal{L}_{\text{pre}} = \frac{1}{|\mathcal{H}|}\sum_{q\in\mathcal{H}}\big(\pi(D(q,F_b)) - m_b(q)\big)^2$, taken separately over the map and the series queries and added. After pretraining, $E$ and $D$ are frozen, $\pi$ is set aside, and the encoder reads all tokens. For a station $s$ in basin $b(s)$ we query once per variable and average the features:

$$r_{s,t} = \frac{1}{|\mathcal{V}|}\sum_{v\in\mathcal{V}} D\big((p_s, \Delta_s, t, v),\, F_{b(s)}\big) \in \mathbb{R}^{d_r}. \tag{1}$$

This is the aligned representation $r_s \in \mathbb{R}^{T\times d_r}$ of Section 3.2. The cell matches what the target measures: the whole basin window for streamflow, the sensor's pixel for stream temperature, snow water equivalent and snow depth. Details are in Appendix E.

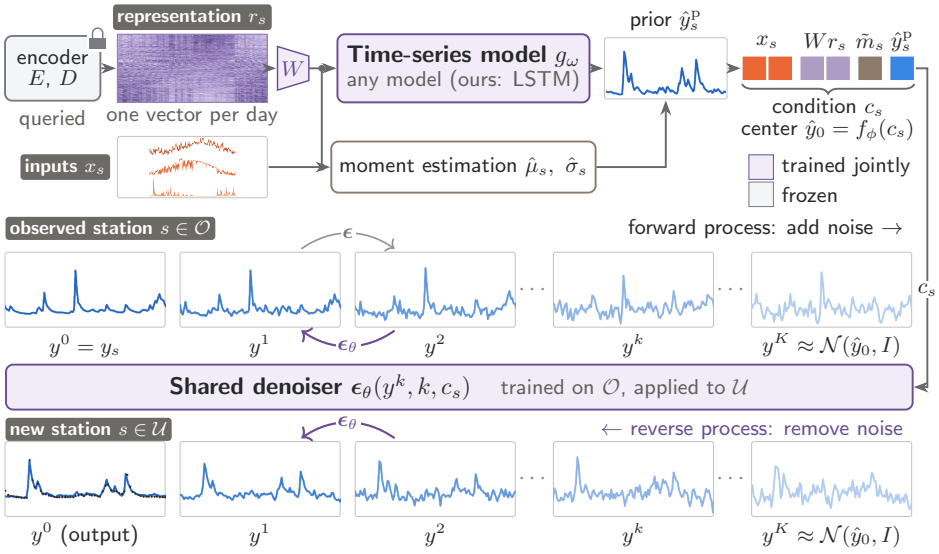


Figure 2: Zero-shot reconstruction on a downstream task (Sections 4.2 and 4.3). Querying the frozen encoder of Figure 1 at a station gives the representation $r_s$, one vector per day. A time-series model $g_\omega$ reads $x_s$ and $Wr_s$ and, with the estimated moments, gives the prior $\hat{y}^{\mathrm{p}}_s$. The condition $c_s$ holds the inputs, $Wr_s$, the moments and the prior, and the diffusion is centered on $\hat{y}_0 = f_\phi(c_s)$: the forward process turns the target into noise around $\hat{y}_0$, the reverse process removes it. One denoiser $\epsilon_\theta$ is trained on $\mathcal{O}$ and applied to $\mathcal{U}$. The curves are streamflow from January to April 2012 at an observed station (target and training row) and at a held-out station (prior and new-station row, dots are observations); the noisy steps are drawn for illustration.

### 4.2 The prior: a time-series model with the representation

We concatenate the representation to the daily inputs of a time-series model $g_\omega$ after a linear projection $W \in \mathbb{R}^{d \times d_r}$, and rescale its output with the moments of the target:

$$\hat{y}^{\mathrm{p}}_s \;=\; \hat{\mu}_s + \hat{\sigma}_s \, g_\omega\big(x_s,\; Wr_s\big). \tag{2}$$

Any model that reads daily inputs can take the role of $g_\omega$ with no change to its design. The width $d$ is chosen for each task on the development folds, and $d = d_r$ means no projection. As in ZeroDiff (Fan et al., 2026), the moments $(\hat{\mu}_s, \hat{\sigma}_s)$ are predicted from the mean and standard deviation of each input column by a conditional VAE $h_\psi$ (Kingma & Welling, 2014; Sohn et al., 2015), so they exist at every station, and $g_\omega$ is trained on $\mathcal{O}$ to fit the standardized target $(y_s - \hat{\mu}_s)/\hat{\sigma}_s$. The predicted moments are used at observed stations too, so the prior makes the same kind of error on both sets. For the two snow tasks the VAE reads station-level inputs rather than basin averages (Section 5.1). Our main $g_\omega$ is an LSTM (Hochreiter & Schmidhuber, 1997); in Section 5.2 published models take its place as they are. Any such model was not built for the representation and may use it poorly or be misled.

### 4.3 Calibration conditioned on the representation

The error of direct fusion, $y_s - \hat{y}^{\mathrm{p}}_s$, is not noise. For a given model, task and representation it follows patterns across stations, and the representation that the prior read is also what describes them. The calibration learns on $\mathcal{O}$ the distribution of the target given the prior and what the prior read,

$$p_\theta\big(y_s \mid c_s\big), \qquad c_s = \big(x_s,\; Wr_s,\; \tilde{m}_s,\; \hat{y}^{\mathrm{p}}_s\big), \tag{3}$$

where $\tilde{m}_s = \big((\hat{\mu}_s - m)/s,\; \hat{\sigma}_s/s\big)$ are the predicted moments in the units of the diffusion, which runs on the target standardized by its mean $m$ and standard deviation $s$ over $\mathcal{O}$. The representation enters on its own and not only through the prior: two stations can share a prior but not its error, and only $Wr_s$ lets the calibration tell them apart. Every entry of $c_s$ exists at an unobserved station, so the distribution learned on $\mathcal{O}$ can be sampled on $\mathcal{U}$.

**A point correction and a spread.** We build equation 3 from two parts that are trained together. A regression network $f_\phi$ (Liu et al., 2022) reads $c_s$ and corrects the prior into a point estimate

$\hat{y}_0 = f_\phi(c_s)$. A diffusion model then describes the spread of the target around $\hat{y}_0$, and it starts from $\hat{y}_0$ rather than from noise (Section 3.3). With $y^0 = y_s$ and a schedule $\beta_k$, $\alpha_k = 1 - \beta_k$, $\bar{\alpha}_k = \prod_{i \le k} \alpha_i$, $\bar{\beta}_k = 1 - \bar{\alpha}_k$, the forward process replaces the target by the estimate and adds noise,

$$y^k \;=\; \sqrt{\bar{\alpha}_k}\, y^0 + \left(1 - \sqrt{\bar{\alpha}_k}\right) \hat{y}_0 + \sqrt{\bar{\beta}_k}\, \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I), \quad k = 1, \ldots, K. \tag{4}$$

The reverse process undoes it with a denoiser $\boldsymbol{\epsilon}_\theta(y^k, k, c_s)$ that reads the whole year of $c_s$ at once and returns an estimate $\epsilon_\theta$ of the noise and a variance $\sigma_\theta$ for the step, as in NsDiff (Ye et al., 2025). Solving equation 4 for the target with $\epsilon_\theta$ in place of $\epsilon$ gives $y^0_\theta = \left(y^k - (1 - \sqrt{\bar{\alpha}_k})\, \hat{y}_0 - \sqrt{\bar{\beta}_k}\, \epsilon_\theta\right) / \sqrt{\bar{\alpha}_k}$, and one step of the reverse process is

$$y^{k-1} \;=\; \gamma_0\, y^0_\theta + \gamma_1\, y^k + \gamma_2\, \hat{y}_0 + \sqrt{\sigma_\theta}\, z, \qquad z \sim \mathcal{N}(0, I), \tag{5}$$

the posterior mean of equation 4 with coefficients that sum to one (Appendix B), plus noise of the learned variance. The estimate $\hat{y}_0$ anchors the chain at its start $y^K = \hat{y}_0 + z$, inside $y^0_\theta$, and in every step, so the denoiser can only move the estimate, and what it learns is how far the target departs from it as a function of $c_s$. Running the chain with fresh noise samples the distribution. Running it with $z = 0$ gives one series for point scores.

**Training.** A station contributes, on its observed days, the estimate's squared error $(\hat{y}_0 - y_s)^2$, the denoising error $(\epsilon - \epsilon_\theta)^2$, and the term $\tilde{\sigma}_{k-1}/\sigma_\theta - \log(\tilde{\sigma}_{k-1}/\sigma_\theta)$ that fits $\sigma_\theta$ to the posterior variance $\tilde{\sigma}_{k-1}$ of equation 4, for random $k$ and $\epsilon$. As in ZeroDiff, stations are weighted by the closeness of $\tilde{m}_s$ to the moments of the unobserved stations. The prior is trained first on its own loss and can then keep its weights or keep training on the calibration loss at a hundredth of the learning rate (Table 2).

# 5 Experiments

## 5.1 Datasets

**Tasks.** We reconstruct four daily targets that are measured at few stations but needed everywhere: streamflow and stream temperature at USGS gauges, snow water equivalent (SWE) and snow depth at SNOTEL stations. Each station belongs to the GAGES-II watershed that contains it, and days without an observation contribute no loss.

**Inputs.** Every model reads the same 38 columns: the 11 basin-averaged NLDAS-2 forcings as daily means and 27 static basin attributes. The gridded imagery and the full table of 605 attributes are seen only through the pretrained representation. The two snow tasks replace one attribute, the fraction of precipitation falling as snow, with the elevation of the station, and their moment estimator reads the five Daymet variables at the station's pixel and that elevation (Section 4.2).

**Folds.** Stations are split into ten spatially contiguous folds, and each fold is held out in turn. Folds 8–10 only choose hyper-parameters; folds 1–7 give every reported result. Details are in Appendix D.

## 5.2 Baselines

We test whether RepCal helps time-series models that were not built for the representation. Published models take the place of $g_\omega$ in equation 2 with their designs unchanged: CNP, TNP and TE-TNP (Garnelo et al., 2018; Nguyen & Grover, 2022; Ashman et al., 2024); ConvCNP and Gridded TNP (Gordon et al., 2020; Ashman et al., 2025); SADI (Islam et al., 2025); and GiFlow and RP Flow (Zhang et al., 2026b; Lalanne et al., 2026). The ZeroDiff row (Fan et al., 2026) is the prior of Section 4. Each model is run three ways: on its own inputs (base), with the representation concatenated to those inputs (+rep, i.e., direct fusion), and with the calibrator of Section 4.3 trained on the frozen +rep prior (+RepCal). The published models are imputers that read same-day observations at neighboring training stations, which ZeroDiff never sees; the calibrator on top of them uses the moment estimator of Section 4.2. Priors at training stations are predicted with each training fold masked in turn, so that no station is predicted by a model that conditioned on its own observations.

Table 1 reports station-median MAE and $R^2$. Direct fusion is inconsistent. By MAE, +rep improves on base for most models on streamflow and stream temperature, but degrades nearly all neural-process and gridded models, as well as GiFlow, on the two snow tasks. +RepCal improves on

Table 1: Station-median MAE and $R^2$ on held-out stations of folds 1–7. base: the model's own inputs; +rep: direct fusion; +RepCal: calibrator on the +rep prior. Bold and shaded: best of the three per model and task. The groups are neural processes, gridded models, and diffusion or flow models.

| | Streamflow | | | Stream temp. | | | SWE | | | Snow depth | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Model | base | +rep | +RepCal | base | +rep | +RepCal | base | +rep | +RepCal | base | +rep | +RepCal |
| **MAE ↓** | | | | | | | | | | | | |
| CNP (2018) | 0.67 | 0.66 | **0.63** | 1.87 | 1.57 | **1.54** | 52.4 | 54.0 | **49.5** | 141.7 | 147.7 | **132.2** |
| TNP (2022) | **0.66** | 0.71 | **0.66** | 1.99 | 1.57 | **1.55** | 51.8 | 57.1 | **50.3** | 139.7 | 138.5 | **134.8** |
| TE-TNP (2024) | 0.64 | 0.66 | **0.62** | 1.98 | 1.51 | **1.50** | **46.7** | 47.9 | 50.2 | **140.2** | 140.4 | 140.4 |
| ConvCNP (2020) | 0.74 | 0.71 | **0.69** | 1.84 | **1.46** | **1.46** | 54.1 | 57.7 | **51.2** | 147.2 | 159.0 | **145.0** |
| Gridded TNP (2025) | 0.72 | 0.67 | **0.66** | 1.69 | 1.49 | **1.46** | 53.8 | 56.3 | **50.4** | **141.1** | 152.7 | 141.2 |
| SADI (2025) | 0.67 | 0.63 | **0.61** | 2.12 | 1.83 | **1.57** | 65.8 | 60.8 | **56.2** | 226.4 | 218.5 | **153.3** |
| GiFlow (2026) | 0.78 | 0.73 | **0.70** | 1.88 | 1.72 | **1.63** | 58.1 | 69.7 | **57.7** | **149.1** | 170.7 | 164.7 |
| RP Flow (2026) | 0.74 | 0.69 | **0.67** | 2.06 | 1.78 | **1.71** | 62.0 | 57.0 | **56.6** | 193.3 | 168.6 | **168.3** |
| ZeroDiff (2026) | 0.79 | 0.80 | **0.70** | 1.88 | 1.90 | **1.79** | 52.5 | 50.3 | **47.1** | 153.0 | **141.3** | 148.0 |
| $\boldsymbol{R^2}$ **↑** | | | | | | | | | | | | |
| CNP (2018) | 0.557 | 0.630 | **0.665** | 0.948 | 0.946 | **0.949** | **0.873** | 0.844 | 0.844 | **0.903** | 0.882 | 0.897 |
| TNP (2022) | 0.549 | 0.606 | **0.674** | 0.937 | 0.948 | **0.952** | **0.868** | 0.832 | 0.846 | **0.900** | 0.888 | 0.888 |
| TE-TNP (2024) | 0.609 | 0.648 | **0.667** | 0.946 | 0.951 | **0.953** | **0.882** | 0.877 | 0.862 | **0.909** | 0.907 | 0.908 |
| ConvCNP (2020) | 0.516 | 0.612 | **0.659** | 0.942 | 0.950 | **0.951** | 0.828 | 0.817 | **0.848** | 0.873 | 0.870 | **0.892** |
| Gridded TNP (2025) | 0.558 | 0.617 | **0.668** | 0.950 | 0.953 | **0.957** | 0.857 | 0.844 | **0.862** | **0.898** | 0.890 | 0.887 |
| SADI (2025) | 0.645 | **0.710** | 0.697 | 0.928 | 0.950 | **0.956** | 0.777 | 0.820 | **0.849** | 0.717 | 0.828 | **0.879** |
| GiFlow (2026) | 0.550 | 0.594 | **0.631** | 0.920 | 0.925 | **0.945** | 0.825 | 0.829 | **0.849** | 0.875 | 0.874 | **0.892** |
| RP Flow (2026) | 0.568 | 0.652 | **0.669** | 0.919 | 0.930 | **0.933** | 0.818 | **0.831** | **0.831** | 0.862 | 0.879 | **0.882** |
| ZeroDiff (2026) | 0.605 | 0.644 | **0.655** | 0.928 | 0.929 | **0.951** | 0.820 | 0.839 | **0.858** | 0.868 | 0.889 | **0.901** |

Table 2: Components of the pipeline on held-out stations of folds 1–7: station-median MAE (left) and $R^2$ (right). Prior: the LSTM on its own inputs; +rep: the representation concatenated to its inputs; +cal.: the calibrator on that prior, with the prior frozen (the +RepCal of Table 1) or trained jointly with it. Bold and shaded: best per task.

| | MAE ↓ | | | | $R^2$ ↑ | | | |
|---|---|---|---|---|---|---|---|---|
| | Streamflow | Stream temp. | SWE | Snow depth | Streamflow | Stream temp. | SWE | Snow depth |
| Prior | 0.79 | 1.88 | 52.5 | 153.0 | 0.605 | 0.928 | 0.820 | 0.868 |
| +rep | 0.80 | 1.90 | 50.3 | 141.3 | 0.644 | 0.929 | 0.839 | 0.889 |
| +cal., joint | **0.67** | 1.97 | 48.6 | **135.6** | 0.656 | 0.943 | 0.847 | 0.884 |
| +rep +cal., prior frozen | 0.70 | **1.79** | **47.1** | 148.0 | 0.655 | **0.951** | **0.858** | 0.901 |
| +rep +cal., joint | 0.68 | 1.80 | 47.6 | 135.7 | **0.687** | 0.945 | 0.849 | **0.906** |

+rep in 32 of the 36 model–task pairs and is the best of the three conditions in 31 (ties included), recovering most of the losses caused by direct fusion. This holds fold by fold: +RepCal beats +rep for a majority of the nine models in 27 of the 28 task–folds. Its gain is largest where the +rep prior is weakest, as for SADI on snow depth (MAE from 218.5 to 153.3), and smallest where that prior is already strong, as for TE-TNP on the snow tasks.

**Ablation.** Table 2 takes the pipeline apart on the LSTM prior, one part at a time: the representation, concatenated to the prior's inputs, and the calibrator, trained on the prior's output. The calibrator can be trained in two ways. With the prior *frozen*, the prior is trained once on its own loss and the calibrator only corrects its output. Trained *jointly*, the prior keeps learning from the calibrator's loss, so the calibrator can also change how the prior reads the representation. The question is whether correcting the output is enough. The representation improves the shape of the series: $R^2$ rises at six or seven of the seven folds on every task, by $0.004$ to $0.030$ in the paired median. The calibrator reduces the size of the error: on the +rep prior it lowers the MAE at every fold on streamflow, stream temperature and SWE. Joint training adds nothing to this. Against the frozen prior it moves the paired median of $R^2$ by at most $0.008$, and its effect on MAE has no consistent sign. Representation and calibrator together improve on the prior in $R^2$ at every fold of every task, and in MAE at 25 of the 28. Correcting the output is enough; the calibrator does not need to change the model it corrects.

**Other priors.** Table 1 changes the model and, with it, the inputs the model reads. Table 3 changes only the prior: the LSTM of Section 4, EA-LSTM (Kratzert et al., 2019b), which reads the static attributes through its input gate, and DLinear (Zeng et al., 2023), a linear model, with the same inputs, representation, moment estimator and calibrator. Direct fusion depends on the prior: the LSTM absorbs the representation, EA-LSTM and DLinear do not, and on stream temperature and the snow tasks it leaves them unchanged or worse, EA-LSTM on SWE at every fold. The calibrator does not: on the frozen +rep prior it improves $R^2$ and MAE for every prior on every task, at five to seven of the seven folds, most where the prior is weakest (DLinear on streamflow, $R^2$ from $0.36$ to $0.64$ at 98% of the stations). On streamflow the spread between the three priors falls from $0.27$ to

Table 3: Three priors with the same inputs and representation, on held-out stations of folds 1–7: station-median MAE (upper panel) and $R^2$ (lower panel). Conditions as in Tables 1 and 2. Bold and shaded: best of the four per prior and task.

| | Streamflow | | | | Stream temp. | | | | SWE | | | | Snow depth | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | +RepCal | | | | +RepCal | | | | +RepCal | | | | +RepCal |
| Prior | base | +rep | frozen | joint | base | +rep | frozen | joint | base | +rep | frozen | joint | base | +rep | frozen | joint |
| **MAE ↓** | | | | | | | | | | | | | | | | |
| LSTM | 0.79 | 0.80 | 0.70 | **0.68** | 1.88 | 1.90 | **1.79** | 1.80 | 52.5 | 50.3 | **47.1** | 47.6 | 153.0 | 141.3 | 148.0 | **135.7** |
| EA-LSTM | 0.79 | 0.81 | 0.70 | **0.68** | 1.76 | 1.82 | 1.75 | **1.72** | **53.5** | 55.0 | 53.8 | 53.8 | 153.0 | 163.4 | **138.9** | 144.7 |
| DLinear | 1.01 | 1.01 | 0.73 | **0.72** | 1.85 | 2.05 | **1.84** | 1.90 | 57.3 | 55.8 | **47.9** | 49.6 | 164.1 | 192.8 | **153.0** | 155.8 |
| **$R^2$ ↑** | | | | | | | | | | | | | | | | |
| LSTM | 0.605 | 0.644 | 0.655 | **0.687** | 0.928 | 0.929 | **0.951** | 0.945 | 0.820 | 0.839 | **0.858** | 0.849 | 0.868 | 0.889 | 0.901 | **0.906** |
| EA-LSTM | 0.560 | 0.617 | **0.666** | 0.661 | 0.937 | 0.930 | **0.945** | 0.941 | 0.815 | 0.750 | 0.835 | **0.839** | 0.864 | 0.849 | 0.880 | **0.885** |
| DLinear | 0.340 | 0.362 | **0.641** | 0.628 | 0.930 | 0.920 | **0.946** | 0.936 | 0.804 | 0.805 | **0.854** | 0.828 | 0.861 | 0.839 | 0.862 | **0.865** |

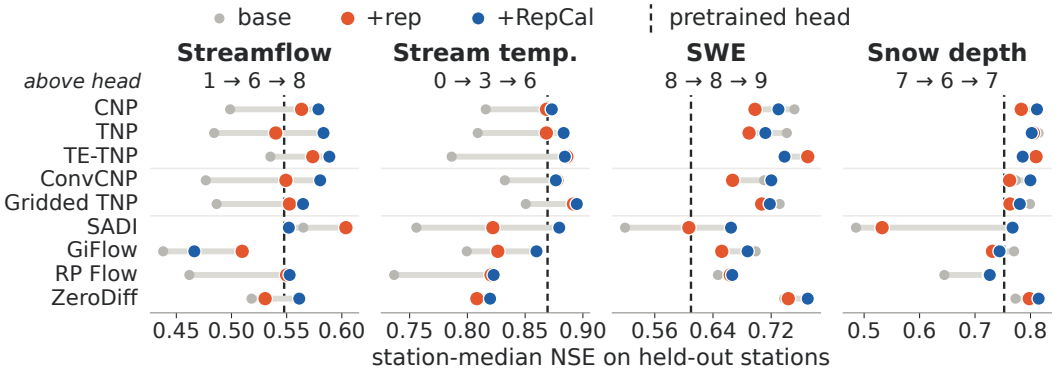


Figure 3: The pretrained model's own head as a reference. Station-median NSE on held-out stations for the nine models of Table 1: base (gray), +rep (orange) and +RepCal (blue). Dashed: the pretrained model's own query decoder trained on the training stations. Numbers under each task: models above the head under base, +rep and +RepCal.

0.03 in $R^2$. Joint training again changes the medians by at most 0.016. With Table 2: the calibrator adapts at the output, so it needs neither to change the prior nor a strong prior.

**Against the pretrained model's representation.** The native way to use a pretrained model is to train a head for its representation. We train the model's own query decoder, 7M of its 34M parameters, on the training stations and score it at the held-out ones. Figure 3 draws it as a line through the nine models, in NSE, the squared error normalized by the station's variance, which scores shape and magnitude together. On streamflow and stream temperature the head is a strong reference: on their own inputs one model passes it on streamflow and none on stream temperature; with the representation six and three do; with the calibrator eight and six. On the two snow tasks, whose target accumulates over months, the head is weak, and seven or eight models pass it on their own inputs. Not passing it with the calibrator are GiFlow on streamflow; GiFlow, RP Flow and ZeroDiff on stream temperature; and GiFlow and RP Flow on snow depth.

## 6 Conclusion

Concatenating a pretrained multimodal representation to the inputs of a time-series model helps on some tasks and hurts on others, but its errors are systematic across stations, and the representation describes them. RepCal therefore reads the representation a second time, as the condition of a diffusion model centered on the fused model's estimate, and calibrates any time-series model without changing it or the frozen encoder. On four environmental targets and several model families, this recovers most of what direct fusion loses and improves on it in most model–task pairs.

### AI USE STATEMENT

In this work, we used generative AI tools to help implement our method and the baselines, by writing and debugging code. We have not used generative AI tools to develop the method, propose hypotheses, design or review the experiments, translate text, or interpret the results, and the remaining tasks with required disclosure (generating synthetic data sets, formulating or proving mathematical claims, and qualitative data analysis) are not applicable to this work. We also used generative AI tools to edit the paper for readability and to format tables in LaTeX. We have reviewed all AI-assisted work: all AI-assisted code was reviewed and tested by the authors before it produced any result, and all AI-edited text was checked against our results. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

### ETHICS STATEMENT

This work uses only publicly available environmental data: gridded weather and static maps, basin attributes, and station observations of streamflow, stream temperature, snow water equivalent and snow depth from public monitoring networks. It involves no human subjects and no personal data. We do not redistribute these data; our repository provides references to the original sources, whose terms of use apply. Reconstructions at stations without observations carry errors that vary across tasks and regions, so they are meant to complement observation-based assessment, not to replace it in decisions such as flood warning or water-supply management.

### REPRODUCIBILITY STATEMENT

The anonymous repository linked in the abstract contains the code to pretrain the encoder and to run all experiments, the weights of the pretrained encoder, and the scripts that download and preprocess the data. The method is described in the method section, and the datasets, splits and evaluation in the experiments section; further details of the pretraining, the adaptations of the baselines, and all hyperparameters and training settings are given in the appendix.

## REFERENCES

Michael Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209):1–80, 2025.

Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. *Gradient Flows in Metric Spaces and in the Space of Probability Measures*. Birkhäuser, 2nd edition, 2008.

Brian D. O. Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12:313–326, 1982.

Matthew Ashman, Cristiana Diaconu, Junhyuck Kim, Lakee Sivaraya, Stratis Markou, James Requeima, Wessel P. Bruinsma, and Richard E. Turner. Translation equivariant transformer neural processes. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, 2024.

Matthew Ashman, Cristiana Diaconu, Eric Langezaal, Adrian Weller, and Richard E. Turner. Gridded transformer neural processes for spatio-temporal data. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025.

Christopher F. Brown, Michal R. Kazmierski, Valerie J. Pasquarella, William J. Rucklidge, Masha Samsikova, Chenhui Zhang, Evan Shelhamer, Estefania Lahera, Olivia Wiles, Simon Ilyushchenko, Noel Gorelick, Lihui Lydia Zhang, Sophia Alj, Emily Schechter, Sean Askay, Oliver Guinan, Rebecca Moore, Alexis Boukouvalas, and Pushmeet Kohli. AlphaEarth Foundations: An embedding field model for accurate and efficient global mapping from sparse label data. arXiv preprint arXiv:2507.22291, 2025.

Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function. In *CVPR*, 2021.

Qi Cheng, Yingda Fan, Xiaowei Jia, Yiqun Xie, and Dan Lu. Hydrotft: a cross-basin attention model for multi-horizon rainfall–runoff prediction. *Machine Learning: Earth*, 2(2):025014, sep 2026. doi: 10.1088/3049-4753/aea164. URL https://doi.org/10.1088/3049-4753/aea164.

Yezhen Cong, Samar Khanna, Chenlin Meng, Patrick Liu, Erik Rozi, Yutong He, Marshall Burke, David Lobell, and Stefano Ermon. SatMAE: Pre-training transformers for temporal and multi-spectral satellite imagery. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

James A. Falcone. GAGES-II: Geospatial attributes of gages for evaluating streamflow. U.S. Geological Survey digital spatial data set, 2011.

Yingda Fan, Dan Lu, and Xiaowei Jia. ZeroDiff: Zero-shot time series reconstruction via informed-prior diffusion. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026. URL https://openreview.net/forum?id=5EKo3JdcR5.

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 *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1704–1713, 2018.

Dean Gesch, Michael Oimoen, Susan Greenlee, Charles Nelson, Michael Steuck, and Dean Tyler. The national elevation data set. *Photogrammetric Engineering and Remote Sensing*, 68(1):5–11, 2002.

Tom Gleeson, Nils Moosdorf, Jens Hartmann, and L. P. H. van Beek. A glimpse beneath earth's surface: GLobal HYdrogeology MaPS (GLHYMPS) of permeability and porosity. *Geophysical Research Letters*, 41(11):3891–3898, 2014. doi: 10.1002/2014GL059856.

Jonathan Gordon, Wessel P. Bruinsma, Andrew Y. K. Foong, James Requeima, Yann Dubois, and Richard E. Turner. Convolutional conditional neural processes. In *International Conference on Learning Representations*, 2020.

Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. In *NeurIPS*, 2021.

Xizewen Han, Huangjie Zheng, and Mingyuan Zhou. CARD: Classification and regression diffusion models. In *NeurIPS*, 2022.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.

Benedikt Heudorfer, Hoshin V. Gupta, and Ralf Loritz. Are deep learning models in hydrology entity aware? *Geophysical Research Letters*, 52:e2024GL113036, 2025. doi: 10.1029/2024GL113036.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

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

M. Hrachowitz, H. H. G. Savenije, G. Blöschl, J. J. McDonnell, M. Sivapalan, J. W. Pomeroy, B. Arheimer, T. Blume, M. P. Clark, U. Ehret, F. Fenicia, J. E. Freer, A. Gelfan, H. V. Gupta, D. A. Hughes, R. W. Hut, A. Montanari, S. Pande, D. Tetzlaff, P. A. Troch, S. Uhlenbrook, T. Wagener, H. C. Winsemius, R. A. Woods, E. Zehe, and C. Cudennec. A decade of predictions in ungauged basins (PUB)—a review. *Hydrological Sciences Journal*, 58(6):1198–1255, 2013. doi: 10.1080/02626667.2013.803183.

Junfeng Hu, Yuxuan Liang, Zhencheng Fan, Hongyang Chen, Yu Zheng, and Roger Zimmermann. Graph neural processes for spatio-temporal extrapolation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023. doi: 10.1145/3580305.3599372.

Jordan Huscroft, Tom Gleeson, Jens Hartmann, and Janine Börker. Compiling and mapping global permeability of the unconsolidated and consolidated Earth: GLobal HYdrogeology MaPS 2.0 (GLHYMPS 2.0). *Geophysical Research Letters*, 45(4):1897–1904, 2018. doi: 10.1002/2017GL075860.

Mohammad Rafid Ul Islam, Prasad Tadepalli, and Alan Fern. Self-attention-based diffusion model for time-series imputation in partial blackout scenarios. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025.

Mohammad Rafid Ul Islam, Prasad Tadepalli, and Alan Fern. Diffusion-based spatio-temporal interpolation with dynamic sensor sets. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, pp. 1968–1977, 2026. doi: 10.1145/3770855.3817811.

Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, Olivier Hénaff, Matthew M. Botvinick, Andrew Zisserman, Oriol Vinyals, and João Carreira. Perceiver IO: A general architecture for structured inputs & outputs. In *ICLR*, 2022.

Yushan Jiang, Kanghui Ning, Zijie Pan, Xuyang Shen, Jingchao Ni, Wenchao Yu, Anderson Schneider, Haifeng Chen, Yuriy Nevmyvaka, and Dongjin Song. Multi-modal time series analysis: A tutorial and survey. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 6043–6053, 2025. doi: 10.1145/3711896.3736567.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. Time-LLM: Time series forecasting by reprogramming large language models. In *International Conference on Learning Representations*, 2024.

Gavin Kerrigan, Giosue Migliorini, and Padhraic Smyth. Functional flow matching. In *Proceedings of the 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pp. 3934–3942, 2024.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun (eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

Konstantin Klemmer, Esther Rolf, Caleb Robinson, Lester Mackey, and Marc Rußwurm. SatCLIP: Global, general-purpose location embeddings with satellite imagery. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 4347–4355, 2025. doi: 10.1609/aaai.v39i4.32457.

Wouter J. M. Knoben, Cyril Thébault, Kasra Keshavarz, Laura Torres-Rojas, Nathaniel W. Chaney, Alain Pietroniro, and Martyn P. Clark. Catchment attributes and MEteorology for large-sample SPATially distributed analysis (CAMELS-SPAT): streamflow observations, forcing data and geospatial data for hydrologic studies across North America. *Hydrology and Earth System Sciences*, 29(20):5791–5833, 2025. doi: 10.5194/hess-29-5791-2025.

Frederik Kratzert, Daniel Klotz, Mathew Herrnegger, Alden K. Sampson, Sepp Hochreiter, and Grey S. Nearing. Toward improved predictions in ungauged basins: Exploiting the power of machine learning. *Water Resources Research*, 55(12):11344–11354, 2019a. doi: 10.1029/2019WR026065.

Frederik Kratzert, Daniel Klotz, Guy Shalev, Günter Klambauer, Sepp Hochreiter, and Grey Nearing. Towards learning universal, regional, and local hydrological behaviors via machine learning applied to large-sample datasets. *Hydrology and Earth System Sciences*, 23:5089–5110, 2019b. doi: 10.5194/hess-23-5089-2019.

Frederik Kratzert, Daniel Klotz, Sepp Hochreiter, and Grey S. Nearing. A note on leveraging synergy in multiple meteorological data sets with deep learning for rainfall–runoff modeling. *Hydrology and Earth System Sciences*, 25(5):2685–2703, 2021. doi: 10.5194/hess-25-2685-2021.

Dohyun Kwon, Ying Fan, and Kangwook Lee. Score-based generative modeling secretly minimizes the Wasserstein distance. In *Advances in Neural Information Processing Systems*, 2022.

Julien Lalanne, David Picard, Lionel Boillot, Lina-María Guayacán-Carrillo, Leon Barens, and Jean-Michel Pereira. Random process flow matching: Generative implicit representations of multivariate random fields. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Zhonghang Li, Lianghao Xia, Jiabin Tang, Yong Xu, Lei Shi, Long Xia, Dawei Yin, and Chao Huang. UrbanGPT: Spatio-temporal large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024. doi: 10.1145/3637528.3671578.

Bryan Lim, Sercan Ö. Arık, 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.

Simon Linke, Bernhard Lehner, Camille Ouellet Dallaire, Joseph Ariwi, Günther Grill, Mira Anand, Penny Beames, Vicente Burchard-Levine, Sally Maxwell, Hana Moidu, Florence Tan, and Michele Thieme. Global hydro-environmental sub-basin and river reach characteristics at high spatial resolution. *Scientific Data*, 6:283, 2019. doi: 10.1038/s41597-019-0300-6.

Chenxi Liu, Qianxiong Xu, Hao Miao, Sun Yang, Lingzheng Zhang, Cheng Long, Ziyue Li, and Rui Zhao. TimeCMA: Towards LLM-empowered multivariate time series forecasting via cross-modality alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 2025. doi: 10.1609/aaai.v39i18.34067.

Haoxin Liu, Shangqing Xu, Zhiyuan Zhao, Lingkai Kong, Harshavardhan Prabhakar Kamarthi, Aditya Sasanur, Megha Sharma, Jiaming Cui, Qingsong Wen, Chao Zhang, and B. Aditya Prakash. Time-MMD: Multi-domain multimodal dataset for time series analysis. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, volume 37, 2024a.

Mingzhe Liu, Han Huang, Hao Feng, Leilei Sun, Bowen Du, and Yanjie Fu. PriSTI: A conditional diffusion framework for spatiotemporal imputation. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pp. 1927–1939, 2023.

Xu Liu, Junfeng Hu, Yuan Li, Shizhe Diao, Yuxuan Liang, Bryan Hooi, and Roger Zimmermann. UniTime: A language-empowered unified model for cross-domain time series forecasting. In *Proceedings of the ACM Web Conference 2024*, 2024b. doi: 10.1145/3589334.3645434.

Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring the stationarity in time series forecasting. In *Advances in Neural Information Processing Systems*, 2022.

Tung Nguyen and Aditya Grover. Transformer neural processes: Uncertainty-aware meta learning via sequence modeling. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 16569–16594, 2022.

Tung Nguyen, Johannes Brandstetter, Ashish Kapoor, Jayesh K. Gupta, and Aditya Grover. ClimaX: A foundation model for weather and climate. In *ICML*, 2023.

Laura Poggio, Luis M. de Sousa, Niels H. Batjes, Gerard B. M. Heuvelink, Bas Kempen, Eloi Ribeiro, and David Rossiter. SoilGrids 2.0: producing soil information for the globe with quantified spatial uncertainty. *SOIL*, 7(1):217–240, 2021. doi: 10.5194/soil-7-217-2021.

Pengfei Qu, Lei Ye, Wenyu Ouyang, Chi Zhang, Yikai Chai, Shuolong Xu, Yongri Piao, Miao Zhang, and Huchuan Lu. Utilizing Earth foundation models to enhance the simulation performance of hydrological models with AlphaEarth embeddings. arXiv preprint arXiv:2601.01558v2, 2026.

Colorado J. Reed, Ritwik Gupta, Shufan Li, Sarah Brockman, Christopher Funk, Brian Clipp, Kurt Keutzer, Salvatore Candido, Matt Uyttendaele, and Trevor Darrell. Scale-MAE: A scale-aware masked autoencoder for multiscale geospatial representation learning. In *ICCV*, 2023.

Esther Rolf, Jonathan Proctor, Tamma Carleton, Ian Bolliger, Vaishaal Shankar, Miyabi Ishihara, Benjamin Recht, and Solomon Hsiang. A generalizable and accessible approach to machine learning with global satellite imagery. *Nature Communications*, 12:4392, 2021. doi: 10.1038/s41467-021-24638-z.

Mehdi S. M. Sajjadi, Henning Meyer, Etienne Pot, Urs Bergmann, Klaus Greff, Noha Radwan, Suhani Vora, Mario Lučić, Daniel Duckworth, Alexey Dosovitskiy, Jakob Uszkoreit, Thomas Funkhouser, and Andrea Tagliasacchi. Scene representation transformer: Geometry-free novel view synthesis through set-latent scene representations. In *CVPR*, 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.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *NeurIPS*, 2015.

Muhammad Talha, A. Pouyan Nejadhashemi, Xiaofeng Liu, and Yuan Yang. BASINGRID: A distributed 40-year spatiotemporal hydrology dataset for U.S. basins. *Scientific Data*, 2026. doi: 10.1038/s41597-026-08030-4.

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, 2021.

Kshitij Tayal, Arvind Renganathan, and Dan Lu. Improving streamflow predictions across conus by integrating advanced machine learning models and diverse data. *Environmental Research Letters*, 19(10):104009, sep 2024. doi: 10.1088/1748-9326/ad6fb7. URL `https://doi.org/10.1088/1748-9326/ad6fb7`.

Peter E. Thornton, Rupesh Shrestha, Michele Thornton, Shih-Chieh Kao, Yaxing Wei, and Bruce E. Wilson. Gridded daily weather data for North America with comprehensive uncertainty quantification. *Scientific Data*, 8:190, 2021. doi: 10.1038/s41597-021-00973-0.

Vinh Ngoc Tran, Donghui Xu, Tam Van Nguyen, Taeho Kim, and Valeriy Y. Ivanov. CAMELSH: A large-sample hourly hydrometeorological dataset and attributes at watershed-scale for CONUS. *Scientific Data*, 12:1307, 2025. doi: 10.1038/s41597-025-05612-6.

Gabriel Tseng, Anthony Fuller, Marlena Reil, Henry Herzog, Patrick Beukema, Favyen Bastani, James R. Green, Evan Shelhamer, Hannah Kerner, and David Rolnick. Galileo: Learning global & local features of many remote sensing modalities. In *ICML*, 2025.

U.S. Geological Survey. Annual NLCD Collection 1 Science Products (ver. 1.1, June 2025). U.S. Geological Survey data release, 2024.

U.S. Geological Survey. USGS Water Data for the Nation: U.S. Geological Survey National Water Information System database, 2026. Stream water temperature, parameter code 00010.

USDA Natural Resources Conservation Service, National Water and Climate Center. Air and Water Database (AWDB) REST API: SNOTEL daily snow water equivalent and snow depth. `https://wcc.sc.egov.usda.gov/awdbRestApi/`, 2026. Accessed August 2026.

Prasanth Valayamkunnath, Michael Barlage, Fei Chen, David J. Gochis, and Kristie J. Franz. Mapping of 30-meter resolution tile-drained croplands using a geospatial modeling approach. *Scientific Data*, 7:257, 2020. doi: 10.1038/s41597-020-00596-x.

Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. VideoMAE V2: Scaling video masked autoencoders with dual masking. In *CVPR*, 2023.

Xinlei Wang, Maike Feng, Jing Qiu, Jinjin Gu, and Junhua Zhao. From news to forecast: Integrating event analysis in LLM-based time series forecasting with reflection. In *Advances in Neural Information Processing Systems*, volume 37, 2024a.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Guo Qin, Haoran Zhang, Yong Liu, Yunzhong Qiu, Jianmin Wang, and Mingsheng Long. TimeXer: Empowering transformers for time series forecasting with exogenous variables. In *Advances in Neural Information Processing Systems*, volume 37, 2024b.

Andrew Robert Williams, Arjun Ashok, Étienne Marcotte, Valentina Zantedeschi, Jithendaraa Subramanian, Roland Riachi, James Requeima, Alexandre Lacoste, Irina Rish, Nicolas Chapados, and Alexandre Drouin. Context is key: A benchmark for forecasting with essential textual information. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 66887–66944, 2025.

Yuankai Wu, Dingyi Zhuang, Aurelie Labbe, and Lijun Sun. Inductive graph neural networks for spatiotemporal kriging. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 2021. doi: 10.1609/aaai.v35i5.16575.

Youlong Xia, Kenneth Mitchell, Michael Ek, Justin Sheffield, Brian Cosgrove, Eric Wood, Lifeng Luo, Charles Alonge, Helin Wei, Jesse Meng, Ben Livneh, Dennis Lettenmaier, Victor Koren, Qingyun Duan, Kingtse Mo, Yun Fan, and David Mocko. Continental-scale water and energy flux analysis and validation for the North American Land Data Assimilation System project phase 2 (NLDAS-2): 1. intercomparison and application of model products. *Journal of Geophysical Research: Atmospheres*, 117:D03109, 2012. doi: 10.1029/2011JD016048.

Qianxiong Xu, Cheng Long, Ziyue Li, Sijie Ruan, Rui Zhao, and Zhishuai Li. KITS: Inductive spatio-temporal kriging with increment training strategy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 2025. doi: 10.1609/aaai.v39i12.33412.

Weiwei Ye, Zhuopeng Xu, and Ning Gui. Non-stationary diffusion for probabilistic time series forecasting. In *ICML*, 2025.

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. doi: 10.1609/aaai.v37i9.26317.

Chuhan Zhang, Guillaume Le Moing, Skanda Koppula, Ignacio Rocco, Liliane Momeni, Junyu Xie, Shuyang Sun, Rahul Sukthankar, Joëlle K. Barral, Raia Hadsell, Zoubin Ghahramani, Andrew Zisserman, Junlin Zhang, and Mehdi S. M. Sajjadi. Efficiently reconstructing dynamic scenes one D4RT at a time. In *CVPR*, 2026a.

Zepeng Zhang, Aref Einizade, Jhony H. Giraldo, and Olga Fink. Spatiotemporal imputation with graph-informed flow matching. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026b.

Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, Jianzhong Qi, Chaochao Chen, and Longbiao Chen. INCREASE: Inductive graph representation learning for spatio-temporal kriging. In *Proceedings of the ACM Web Conference 2023*, 2023. doi: 10.1145/3543507.3583525.

Tian Zhou, Peisong Niu, Xue Wang, Liang Sun, and Rong Jin. One fits all: Power general time series analysis by pretrained LM. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

# A ALGORITHM

Algorithms 1 and 2 put the pieces of Sections 4.2 and 4.3 together with the derivations of Appendix B. The encoder $E$ and the decoder $D$ come from Section 4.1 and stay frozen throughout; $\eta$ is the learning rate of the calibration.

**Algorithm 1** Training

**Input:** stations $\mathcal{O}$ with $(x_s, y_s)$ and $\mathcal{U}$ with $x_s$; multimodal inputs $\mathcal{M}_{b(s)}$; frozen $E$, $D$; steps $K$, schedule $\beta_1, \dots, \beta_K$
For all $s \in \mathcal{O} \cup \mathcal{U}$: $r_s \leftarrow$ query $D(\,\cdot\,, E(\mathcal{M}_{b(s)}))$ at the station *// equation 1*
Train the moment model $h_\psi$ on $\mathcal{O}$; for all $s$: $(\hat{\mu}_s, \hat{\sigma}_s) \leftarrow h_\psi(x_s)$ *// moments*
Train $W$ and $g_\omega$ on $\mathcal{O}$ to fit $(y_s - \hat{\mu}_s)/\hat{\sigma}_s$ *// the prior, equation 2*
For all $s$: $\tilde{m}_s$ from $(\hat{\mu}_s, \hat{\sigma}_s)$; $\tilde{m}^* \leftarrow$ mean over $\mathcal{U}$; weights $w_s$, $s \in \mathcal{O}$ *// Section 4.3*
**repeat**
  Take a batch of stations $s \in \mathcal{O}$
  $\hat{y}^{\mathrm{p}}_s \leftarrow \hat{\mu}_s + \hat{\sigma}_s\, g_\omega(x_s, Wr_s)$; $c_s \leftarrow (x_s, Wr_s, \tilde{m}_s, \hat{y}^{\mathrm{p}}_s)$; $\hat{y}_0 \leftarrow f_\phi(c_s)$ *// prior recomputed*
  Sample $k \sim \mathrm{Uniform}\{1, \dots, K\}$ and $\epsilon \sim \mathcal{N}(0, I)$; fill the unobserved days of $y_s$ with $\hat{y}_0$
  $y^k \leftarrow \sqrt{\bar{\alpha}_k}\, y_s + (1 - \sqrt{\bar{\alpha}_k})\, \hat{y}_0 + \sqrt{\bar{\beta}_k}\, \epsilon$ *// equation 7*
  $(\epsilon_\theta, \sigma_\theta) \leftarrow \boldsymbol{\epsilon}_\theta(y^k, k, c_s)$
  $\mathcal{L}_s \leftarrow$ equation 17 over the observed days of $s$
  Update $\theta, \phi$ with rate $\eta$ on $\sum_s w_s \mathcal{L}_s / \sum_s w_s$, and optionally $\omega, W$ with rate $\eta/100$
**until** convergence

**Algorithm 2** Inference at a new station

**Input:** $s \in \mathcal{U}$ with $x_s$ and $\mathcal{M}_{b(s)}$; frozen $E$, $D$; trained $h_\psi$, $W$, $g_\omega$, $f_\phi$, $\boldsymbol{\epsilon}_\theta$
$r_s \leftarrow$ query at the station; $(\hat{\mu}_s, \hat{\sigma}_s) \leftarrow h_\psi(x_s)$; $\hat{y}^{\mathrm{p}}_s \leftarrow \hat{\mu}_s + \hat{\sigma}_s\, g_\omega(x_s, Wr_s)$ *// prior*
$c_s \leftarrow (x_s, Wr_s, \tilde{m}_s, \hat{y}^{\mathrm{p}}_s)$; $\hat{y}_0 \leftarrow f_\phi(c_s)$ *// center*
$y^K \leftarrow \hat{y}_0 + z$, $z \sim \mathcal{N}(0, I)$ *// start around the center*
**for** $k = K$ **to** $1$ **do**
  $(\epsilon_\theta, \sigma_\theta) \leftarrow \boldsymbol{\epsilon}_\theta(y^k, k, c_s)$
  $v \leftarrow$ larger root of $\Sigma_1(v)\,\Sigma_2(v) = \sigma_\theta\big(\alpha_k \Sigma_2(v) + \Sigma_1(v)\big)$ *// equation 20*
  $y^0_\theta \leftarrow \frac{1}{\sqrt{\bar{\alpha}_k}}\Big(y^k - (1 - \sqrt{\bar{\alpha}_k})\, \hat{y}_0 - \sqrt{\bar{\beta}_k + \tilde{\beta}_k(v-1)}\, \epsilon_\theta\Big)$ *// equation 15*
  **if** $k > 1$: $y^{k-1} \leftarrow \gamma_0\, y^0_\theta + \gamma_1\, y^k + \gamma_2\, \hat{y}_0 + \sqrt{\sigma_\theta}\, z$, $z \sim \mathcal{N}(0, I)$ *// equation 18, $\gamma$ at $v$*
**end for**
**Output:** $y^0_\theta$ from the last step. Repeat for samples of the distribution; set every $z = 0$ for the point prediction

# B Bayesian Derivation of the Posterior Distribution

This section derives the forward process, the posterior and the sampling rule of the calibration of Section 4.3, and states how the variance of the reverse process is learned. The derivation follows ZeroDiff (Fan et al., 2026); the learned variance follows NsDiff (Ye et al., 2025). We drop the station index $s$ and write $y^0 \in \mathbb{R}^T$ for the target series, $c$ for the condition, and $\hat{y}_0 = f_\phi(c) \in \mathbb{R}^T$ for the center of the diffusion. Diffusion steps are indexed by $k = 1, \dots, K$, with a noise schedule $\beta_k \in (0,1)$, $\alpha_k = 1 - \beta_k$, $\bar{\alpha}_k = \prod_{i=1}^{k} \alpha_i$ and $\bar{\beta}_k = 1 - \bar{\alpha}_k$. All series are treated day by day: every square, product, quotient and variance below is taken entry by entry over the $T$ days, and $I$ is the identity.

## B.1 Forward process

Standard DDPM (Ho et al., 2020) moves the target toward pure noise, $y^K \sim \mathcal{N}(0, I)$. We instead move it toward the center $\hat{y}_0$, as in CARD (Han et al., 2022). The one-step transition is

$$y^k = \sqrt{\alpha_k}\, y^{k-1} + (1 - \sqrt{\alpha_k})\, \hat{y}_0 + \sqrt{\beta_k}\, \epsilon_k, \qquad \epsilon_k \sim \mathcal{N}(0, I). \tag{6}$$

Unrolling the recursion,

$$\begin{aligned} y^k &= \sqrt{\alpha_k}\Big[\sqrt{\alpha_{k-1}}\, y^{k-2} + (1 - \sqrt{\alpha_{k-1}})\, \hat{y}_0 + \sqrt{\beta_{k-1}}\, \epsilon_{k-1}\Big] + (1 - \sqrt{\alpha_k})\, \hat{y}_0 + \sqrt{\beta_k}\, \epsilon_k \\ &= \sqrt{\alpha_k \alpha_{k-1}}\, y^{k-2} + \big(1 - \sqrt{\alpha_k \alpha_{k-1}}\big)\, \hat{y}_0 + \sqrt{\alpha_k \beta_{k-1} + \beta_k}\, \epsilon \\ &\;\;\vdots \\ &= \sqrt{\bar{\alpha}_k}\, y^0 + (1 - \sqrt{\bar{\alpha}_k})\, \hat{y}_0 + \sqrt{\bar{\beta}_k}\, \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I), \end{aligned} \tag{7}$$

where independent noise terms are merged by adding their variances, and $\alpha_k \beta_{k-1} + \beta_k = 1 - \alpha_k \alpha_{k-1}$. Thus

$$q(y^k \mid y^0, \hat{y}_0) = \mathcal{N}\big(\sqrt{\bar{\alpha}_k}\, y^0 + (1 - \sqrt{\bar{\alpha}_k})\, \hat{y}_0,\ \bar{\beta}_k I\big), \tag{8}$$

which is equation 4 of the main text with $\bar{\beta}_k = 1 - \bar{\alpha}_k$. As $\bar{\alpha}_K \to 0$ the forward process ends at $\mathcal{N}(\hat{y}_0, I)$, a cloud around the center rather than around zero.

## B.2 Reverse posterior

By Bayes' rule,

$$q(y^{k-1} \mid y^k, y^0, \hat{y}_0) \;\propto\; q(y^k \mid y^{k-1}, \hat{y}_0)\; q(y^{k-1} \mid y^0, \hat{y}_0). \tag{9}$$

From equation 6 the transition is $q(y^k \mid y^{k-1}, \hat{y}_0) = \mathcal{N}\big(\sqrt{\alpha_k}\, y^{k-1} + (1 - \sqrt{\alpha_k})\, \hat{y}_0,\ \beta_k I\big)$, and from equation 7 at step $k-1$, $q(y^{k-1} \mid y^0, \hat{y}_0) = \mathcal{N}\big(\sqrt{\bar{\alpha}_{k-1}}\, y^0 + (1 - \sqrt{\bar{\alpha}_{k-1}})\, \hat{y}_0,\ \bar{\beta}_{k-1} I\big)$. Write $A = y^k - (1 - \sqrt{\alpha_k})\, \hat{y}_0$ and $B = \sqrt{\bar{\alpha}_{k-1}}\, y^0 + (1 - \sqrt{\bar{\alpha}_{k-1}})\, \hat{y}_0$. The product of the two Gaussians is

$$\begin{aligned} q(y^{k-1} \mid y^k, y^0, \hat{y}_0) &\propto \exp\Big(-\frac{(A - \sqrt{\alpha_k}\, y^{k-1})^2}{2\beta_k} - \frac{(y^{k-1} - B)^2}{2\bar{\beta}_{k-1}}\Big) \\ &= \exp\Big(-\frac{1}{2}\Big[\Big(\frac{\alpha_k}{\beta_k} + \frac{1}{\bar{\beta}_{k-1}}\Big)(y^{k-1})^2 - 2\Big(\frac{\sqrt{\alpha_k}\, A}{\beta_k} + \frac{B}{\bar{\beta}_{k-1}}\Big) y^{k-1} + \text{const}\Big]\Big). \end{aligned} \tag{10}$$

Completing the square in $y^{k-1}$, the posterior is Gaussian with

$$\tilde{\sigma}_{k-1} = \Big(\frac{\alpha_k}{\beta_k} + \frac{1}{\bar{\beta}_{k-1}}\Big)^{-1} = \frac{\beta_k\, \bar{\beta}_{k-1}}{\alpha_k \bar{\beta}_{k-1} + \beta_k}, \tag{11}$$

$$\tilde{\mu}_{k-1} = \tilde{\sigma}_{k-1}\Big(\frac{\sqrt{\alpha_k}\, A}{\beta_k} + \frac{B}{\bar{\beta}_{k-1}}\Big) = \frac{\sqrt{\alpha_k}\, \bar{\beta}_{k-1}\, A + \beta_k\, B}{\alpha_k \bar{\beta}_{k-1} + \beta_k}. \tag{12}$$

Substituting $A$ and $B$ into equation 12 and collecting the terms in $y^0$, $y^k$ and $\hat{y}_0$,

$$\begin{aligned}\tilde{\mu}_{k-1} &= \frac{\sqrt{\alpha_k}\,\bar{\beta}_{k-1}}{\alpha_k\bar{\beta}_{k-1}+\beta_k}\big[y^k-(1-\sqrt{\alpha_k})\,\hat{y}_0\big] \\ &\quad+\frac{\beta_k}{\alpha_k\bar{\beta}_{k-1}+\beta_k}\big[\sqrt{\bar{\alpha}_{k-1}}\,y^0+(1-\sqrt{\bar{\alpha}_{k-1}})\,\hat{y}_0\big] \\ &= \gamma_0\,y^0+\gamma_1\,y^k+\gamma_2\,\hat{y}_0,\end{aligned} \tag{13}$$

with

$$\begin{aligned}\gamma_0 &= \frac{\sqrt{\bar{\alpha}_{k-1}}\,\beta_k}{\alpha_k\bar{\beta}_{k-1}+\beta_k}, \qquad \gamma_1 = \frac{\sqrt{\alpha_k}\,\bar{\beta}_{k-1}}{\alpha_k\bar{\beta}_{k-1}+\beta_k}, \\ \gamma_2 &= \frac{\sqrt{\alpha_k}\,(\sqrt{\alpha_k}-1)\,\bar{\beta}_{k-1}+(1-\sqrt{\bar{\alpha}_{k-1}})\,\beta_k}{\alpha_k\bar{\beta}_{k-1}+\beta_k}.\end{aligned} \tag{14}$$

The three coefficients sum to one. In standard DDPM the posterior mean has only the first two terms. The third term $\gamma_2\hat{y}_0$ pulls every step toward the center.

### B.3 Learned variance and training loss

The reverse process is a Gaussian $p_\theta(y^{k-1} \mid y^k, \hat{y}_0, c) = \mathcal{N}\big(\mu_\theta,\ \mathrm{diag}(\sigma_\theta)\big)$ whose mean and variance come from one network with two heads, the denoiser $\boldsymbol{\epsilon}_\theta(y^k, k, c)$ of Figure 2. It reads $y^k$, the step $k$, the condition $c$ and the center $\hat{y}_0 = f_\phi(c)$, and returns a noise estimate $\epsilon_\theta \in \mathbb{R}^T$ and a variance $\sigma_\theta \in \mathbb{R}^T_{>0}$, one value per day, as in NsDiff (Ye et al., 2025). The mean $\mu_\theta$ is equation 13 with $y^0$ replaced by the estimate that $\epsilon_\theta$ implies through equation 7,

$$y^0_\theta = \frac{1}{\sqrt{\bar{\alpha}_k}}\Big(y^k-(1-\sqrt{\bar{\alpha}_k})\,\hat{y}_0-\sqrt{\bar{\beta}_k}\,\epsilon_\theta\Big). \tag{15}$$

The Kullback–Leibler divergence from the posterior $\mathcal{N}(\tilde{\mu}_{k-1}, \tilde{\sigma}_{k-1}I)$ to $p_\theta$ is, day by day,

$$\frac{1}{2}\Big[\frac{(\tilde{\mu}_{k-1}-\mu_\theta)^2}{\sigma_\theta}+\frac{\tilde{\sigma}_{k-1}}{\sigma_\theta}-\log\frac{\tilde{\sigma}_{k-1}}{\sigma_\theta}-1\Big], \tag{16}$$

where $\tilde{\mu}_{k-1}-\mu_\theta = \gamma_0\,(y^0-y^0_\theta)$ is proportional to $\epsilon-\epsilon_\theta$. As in DDPM and NsDiff we drop the constant and the step-dependent weight of the first term. A station with observed days $\mathcal{T} \subseteq \{1,\dots,T\}$ then contributes

$$\mathcal{L} = \mathbb{E}_{k,\epsilon}\,\frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}}\Big[(\hat{y}_0-y^0)^2+(\epsilon-\epsilon_\theta)^2+\frac{\tilde{\sigma}_{k-1}}{\sigma_\theta}-\log\frac{\tilde{\sigma}_{k-1}}{\sigma_\theta}\Big]_t, \tag{17}$$

with $k$ uniform on $\{1,\dots,K\}$, $\epsilon \sim \mathcal{N}(0,I)$, $y^k$ from equation 7, and $[\,\cdot\,]_t$ the entry of day $t$. The first term trains the regression network $f_\phi$, the second the noise head, and the last two the variance head; they are smallest at $\sigma_\theta = \tilde{\sigma}_{k-1}$. At $k=1$, where $\tilde{\sigma}_0 = 0$, $\beta_1$ takes its place. Days without an observation contribute nothing to the loss, and their target is replaced by $\hat{y}_0$ before the noise is added, so that the denoiser always reads a full year. As in ZeroDiff, a station is weighted by its closeness to the unobserved stations in the moment coordinates $\tilde{m}$ of Section 4.3, $w_s = 0.1+1.9\exp\big(-\|\tilde{m}_s-\tilde{m}^*\|^2/2\tau^2\big)$, where $\tilde{m}^*$ is the mean of $\tilde{m}$ over $\mathcal{U}$, computed from inputs only, and $\tau = 0.5$ is half a standard deviation of the target, so one value serves every task. The prior is recomputed at every step in training mode, so its input dropout of $0.2$ stays active whether or not its weights are updated: the calibration is trained on a perturbed prior and applied to the deterministic one. If the prior keeps training with the calibration, the gradient of $\mathcal{L}$ also reaches $\omega$ and $W$ through $c$.

### B.4 Sampling

At a new station the chain starts around the center, $y^K = \hat{y}_0 + z$ with $z \sim \mathcal{N}(0,I)$, the terminal distribution of CARD, and for $k = K,\dots,2$ takes the step

$$y^{k-1} = \gamma_0\,y^0_\theta+\gamma_1\,y^k+\gamma_2\,\hat{y}_0+\sqrt{\sigma_\theta}\,z, \qquad z\sim\mathcal{N}(0,I), \tag{18}$$

where $\epsilon_\theta$ and $\sigma_\theta$ are the two outputs of $\boldsymbol{\epsilon}_\theta(y^k, k, c)$. The learned $\sigma_\theta$ takes the place of the schedule's $\tilde{\sigma}_{k-1}$ as the noise of the reverse step. The last step, $k = 1$, returns $y^0_\theta$: there $\bar{\alpha}_0 = 1$ gives $\gamma_0 = 1$, $\gamma_1 = \gamma_2 = 0$ and $\tilde{\sigma}_0 = 0$. The center $\hat{y}_0$ thus enters at three places, the start of the chain, the estimate equation 15, and the term $\gamma_2 \hat{y}_0$ of every step, and the denoiser learns how the target departs from the center rather than the target itself. For a point prediction we set $z = 0$ at the start and at every step, which gives one deterministic trajectory from $\hat{y}_0$. With the schedule of Appendix E ($K = 20$), $\bar{\alpha}_K$ is far from zero and this start is wider than the end point $\bar{\beta}_K I$ of the forward process. The width of the start is a hyperparameter chosen on the development folds, where the unit width was better than the width matched to the forward process, $\sqrt{\bar{\alpha}_K v_r + \bar{\beta}_K}$ with $v_r$ the variance of $y^0 - \hat{y}_0$ over the training stations.

**Reading the target variance off $\sigma_\theta$.** NsDiff pairs the learned variance with a forward process in which the target is allowed a variance $v$ around the center, different from the unit variance that the chain starts from. The transition noise of equation 6 then has variance $\beta_k(\beta_k + \alpha_k v)$ instead of $\beta_k$, and unrolling as in equation 7 gives the marginal variance

$$\bar{\beta}_k + \tilde{\beta}_k\,(v - 1), \qquad \tilde{\beta}_k = \sum_{i=1}^{k} \beta_i \prod_{j=i}^{k} \alpha_j, \tag{19}$$

so the spread of $y^k$ around the center moves from $v$ at $k = 0$ toward one as $k$ grows, and $v = 1$ recovers equation 7. The derivation of Section B.2 goes through with $\beta_k$ replaced by $\Sigma_1(v) = \beta_k(\beta_k + \alpha_k v)$ and $\bar{\beta}_{k-1}$ by $\Sigma_2(v) = \bar{\beta}_{k-1} + \tilde{\beta}_{k-1}(v - 1)$:

$$\tilde{\sigma}_{k-1}(v) = \frac{\Sigma_1 \Sigma_2}{\alpha_k \Sigma_2 + \Sigma_1}, \qquad \gamma_0 = \frac{\sqrt{\bar{\alpha}_{k-1}}\,\Sigma_1}{\alpha_k \Sigma_2 + \Sigma_1}, \qquad \gamma_1 = \frac{\sqrt{\alpha_k}\,\Sigma_2}{\alpha_k \Sigma_2 + \Sigma_1},$$
$$\gamma_2 = \frac{\sqrt{\alpha_k}(\sqrt{\alpha_k} - 1)\,\Sigma_2 + (1 - \sqrt{\bar{\alpha}_{k-1}})\,\Sigma_1}{\alpha_k \Sigma_2 + \Sigma_1}. \tag{20}$$

Training uses $v = 1$. At sampling time, at every step and day, we read the learned $\sigma_\theta$ as the posterior variance $\tilde{\sigma}_{k-1}(v)$ of such a process and solve $\Sigma_1(v)\,\Sigma_2(v) = \sigma_\theta\big(\alpha_k \Sigma_2(v) + \Sigma_1(v)\big)$ for $v$, a quadratic whose larger root is the solution. This $v$ replaces $\bar{\beta}_k$ in equation 15 by $\bar{\beta}_k + \tilde{\beta}_k(v - 1)$ and sets the coefficients of equation 18 through equation 20. For $\sigma_\theta = \tilde{\sigma}_{k-1}$ the root is $v = 1$ and the step is the posterior of Section B.2.

## C A BOUND ON THE CALIBRATED DISTRIBUTION

This section bounds the distance between the distribution that the calibration samples and the true conditional distribution of the target. It follows the Wasserstein analysis of score-based models by Kwon et al. (2022), applied to a forward process that moves toward the center, and it makes precise what Section 4.3 says in words: the error of the samples is at most the error of the center plus a term that a short chain keeps small, and the representation in the condition can only lower the first part. As in Appendix B we fix a station, drop its index, and write $c$ for the condition and $\hat{y}_0 = f_\phi(c)$ for the center.

**Continuous-time process.** Let the step $k$ run continuously over $[0, K]$ with a schedule $\beta_k > 0$, and let $\bar{\alpha}_k = \exp\big(-\int_0^k \beta_u \, du\big)$ and $\bar{\beta}_k = 1 - \bar{\alpha}_k$. The forward process of Appendix B.1 is the small-step limit of

$$dy = -\tfrac{1}{2}\beta_k \, (y - \hat{y}_0) \, dk + \sqrt{\beta_k} \, dw, \tag{21}$$

an Ornstein–Uhlenbeck process that reverts to the center. Its marginal given $y^0$ is $\mathcal{N}\big(\sqrt{\bar{\alpha}_k}\, y^0 + (1 - \sqrt{\bar{\alpha}_k})\, \hat{y}_0,\ \bar{\beta}_k I\big)$, as in equation 7. Let $q_k$ be the density of $y^k$ given $c$ when $y^0$ is drawn from the true conditional distribution $q_0$ of the target. Reversing equation 21 in $k$ (Anderson, 1982) and replacing the true score $\nabla \log q_k$ by a learned score $s_\theta(y, k, c)$ gives, with $k$ decreasing,

$$dy = \Big[-\tfrac{1}{2}\beta_k \, (y - \hat{y}_0) - \beta_k \, s_\theta(y, k, c)\Big] dk + \sqrt{\beta_k} \, d\bar{w}, \qquad dk < 0, \tag{22}$$

started at level $K$ from a distribution $p_K$. Let $p_k$ be the density of this process at level $k$, so that $p_0$ is what the calibration samples. In the sampler of Appendix B.4, $s_\theta = -\epsilon_\theta / \sqrt{\bar{\beta}_k}$.

**Assumption 1** (one-sided Lipschitz score)**.** *There is $l_k \in \mathbb{R}$ such that for all $y_1, y_2$, $\langle s_\theta(y_1, k, c) - s_\theta(y_2, k, c),\ y_1 - y_2 \rangle \le l_k \|y_1 - y_2\|^2$.*

**Assumption 2** (regularity)**.** *For every $k \in [0, K]$ the densities $q_k$ and $p_k$ are positive and twice continuously differentiable, have finite second moments, and the velocity fields of their continuity equations are square integrable under them, so that $k \mapsto q_k$ and $k \mapsto p_k$ are absolutely continuous curves in the Wasserstein space $(\mathcal{P}_2, W_2)$.*

**Proposition 1.** *Let $H_k = \mathbb{E}_{y \sim q_k} \|s_\theta(y, k, c) - \nabla \log q_k(y)\|^2$ be the score error and $M_k = \exp\big(\int_0^k (\tfrac{1}{2} + l_u)\, \beta_u \, du\big)$. Under Assumptions 1 and 2,*

$$W_2(q_0, p_0) \ \le \ \int_0^K \beta_k M_k \sqrt{H_k} \, dk \ + \ M_K \, W_2(q_K, p_K). \tag{23}$$

*If the chain starts at the end point of the forward process, $p_K = \mathcal{N}(\hat{y}_0, \bar{\beta}_K I)$, then*

$$W_2(q_K, p_K) \ \le \ \sqrt{\bar{\alpha}_K} \, \big(\mathbb{E}_{q_0} \|y^0 - \hat{y}_0\|^2\big)^{1/2}. \tag{24}$$

*Proof. Velocity fields.* The Fokker–Planck equation of equation 21 is the continuity equation $\partial_k q_k + \nabla \cdot (q_k v_k) = 0$ with $v_k(y) = -\frac{1}{2}\beta_k (y - \hat{y}_0) - \frac{1}{2}\beta_k \nabla \log q_k(y)$. The process equation 22, read forward in its own time $K - k$ and re-indexed by $k$, has density $p_k$ with $\partial_k p_k + \nabla \cdot (p_k w_k) = 0$ and $w_k(z) = -\frac{1}{2}\beta_k (z - \hat{y}_0) - \beta_k s_\theta(z, k, c) + \frac{1}{2}\beta_k \nabla \log p_k(z)$. When $s_\theta = \nabla \log q_k$ and $p_k = q_k$, $w_k = v_k$: the reverse process reproduces the forward marginals.

*Derivative of the distance.* By Assumption 2 and Theorem 8.4.7 of Ambrosio et al. (2008), for almost every $k$, with $\pi_k$ an optimal coupling of $q_k$ and $p_k$,

$$\frac{d}{dk} \, \tfrac{1}{2} W_2^2(q_k, p_k) = \mathbb{E}_{(y,z) \sim \pi_k} \big[(y - z) \cdot (v_k(y) - w_k(z))\big]. \tag{25}$$

Write $W = W_2(q_k, p_k)$. Inserting the fields, the center cancels and

$$\begin{aligned} -W \frac{dW}{dk} &= \tfrac{1}{2}\beta_k W^2 + \beta_k \, \mathbb{E}_{\pi_k} \big[(y - z) \cdot (\nabla \log q_k(y) - s_\theta(z, k, c))\big] \\ &\quad - \tfrac{1}{2}\beta_k \, \mathbb{E}_{\pi_k} \big[(y - z) \cdot (\nabla \log q_k(y) - \nabla \log p_k(z))\big]. \end{aligned} \tag{26}$$

The last expectation is nonnegative under an optimal coupling, by the convexity of the entropy along generalized geodesics (Lemma 1 of Kwon et al. (2022)), so the last term is at most zero. In

the middle term we add and subtract $s_\theta(y, k, c)$. By Assumption 1, $\mathbb{E}_{\pi_k}[(y - z) \cdot (s_\theta(y, k, c) - s_\theta(z, k, c))] \le l_k W^2$, and by the Cauchy–Schwarz inequality, the first marginal of $\pi_k$ being $q_k$, $\mathbb{E}_{\pi_k}[(y - z) \cdot (\nabla \log q_k(y) - s_\theta(y, k, c))] \le W\sqrt{H_k}$. Dividing by $W > 0$,

$$-\frac{dW}{dk} \;\le\; \big(\tfrac{1}{2} + l_k\big)\beta_k\, W + \beta_k\sqrt{H_k}. \tag{27}$$

*Integration.* Since $\frac{d}{dk}M_k = (\frac{1}{2} + l_k)\beta_k M_k$, equation 27 gives $-\frac{d}{dk}(M_k W) \le \beta_k M_k \sqrt{H_k}$. Integrating from 0 to $K$ with $M_0 = 1$ yields equation 23.

*Start of the chain.* Draw $y^0 \sim q_0$ and $\epsilon \sim \mathcal{N}(0, I)$, and set $y^K = \sqrt{\bar\alpha_K}\, y^0 + (1 - \sqrt{\bar\alpha_K})\, \hat y_0 + \sqrt{\bar\beta_K}\, \epsilon \sim q_K$ and $z^K = \hat y_0 + \sqrt{\bar\beta_K}\, \epsilon \sim p_K$. This couples $q_K$ and $p_K$ with $\|y^K - z^K\|^2 = \bar\alpha_K \|y^0 - \hat y_0\|^2$, and $W_2^2$ is at most the cost of any coupling, which gives equation 24. □

**Corollary 1.** *In the setting of Proposition 1,*

$$W_2(q_0, p_0) \;\le\; \int_0^K \beta_k M_k \sqrt{H_k}\, dk \;+\; M_K \sqrt{\bar\alpha_K}\, \big(\mathbb{E}_{q_0}\|y^0 - f_\phi(c)\|^2\big)^{1/2}. \tag{28}$$

*Moreover, the smallest value of the last expectation over all centers that are functions of the condition does not increase when the condition is enlarged: if $c'$ is a function of $c$, then* $\min_f \mathbb{E}\|y^0 - f(c)\|^2 \le \min_f \mathbb{E}\|y^0 - f(c')\|^2$.

*Proof.* The bound is equation 23 with equation 24. For the second claim, every function of $c'$ is a function of $c$, and the minimum over a larger class of functions is not larger. □

**What the bound says.** With the schedule of Appendix E, $\sqrt{\bar\alpha_K} = 0.95$ and $\int_0^K \beta_k\, dk = 0.10$, so $M_K = \exp(0.05 + 0.10\, l)$ for a constant $l$. The samples of the calibration are then within about the error of the center of the true target, plus a score term that carries the factor $0.10$: a short chain cannot damage a good center by more than a small multiple of its score error. The center is $f_\phi(c)$ with $c = (x,\, Wr,\, \tilde m,\, \hat y^{\mathrm{p}})$. By Corollary 1, the best center with $Wr$ in the condition is at least as good as the best one without it, and at least as good as the prior, which is itself a function of $c$. This is the precise form of the claim in Section 4.3 that the representation enters the condition directly. Finally, if the chain were made long with the same $\beta$, $\sqrt{\bar\alpha_K}$ would go to zero and remove the center from the bound, but $M_K$ would grow exponentially and the score term would accumulate over the whole chain. A short chain around a good center keeps both terms small, which is why the calibration inherits the center rather than generating from noise.

The bound is stated for the continuous-time process; the discrete sampler of Appendix B.4 adds a discretization error that it does not cover. It also takes the chain to start at the end point of the forward process, whereas the width of the start in Appendix B.4 is chosen as a hyperparameter. It holds for each station given its condition; averaging over stations is an expectation of both sides.

Tables 4–6 report the runs of Table 1 under the other metrics. Figure 4 gives, for each model, task and metric, the share of held-out stations where +rep and +RepCal do better than base. Figure 5 shows daily series of SADI at eight of these stations.

Table 4: NSE, station median (higher is better).

| | Streamflow | | | Stream temp. | | | SWE | | | Snow depth | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Model | base | +rep | +RepCal | base | +rep | +RepCal | base | +rep | +RepCal | base | +rep | +RepCal |
| *Neural processes* | | | | | | | | | | | | |
| CNP (2018) | 0.499 | 0.564 | **0.579** | 0.816 | 0.869 | **0.873** | **0.752** | 0.698 | 0.730 | 0.811 | 0.783 | **0.812** |
| TNP (2022) | 0.484 | 0.540 | **0.583** | 0.809 | 0.868 | **0.883** | **0.742** | 0.690 | 0.712 | **0.815** | 0.805 | 0.802 |
| TE-TNP (2024) | 0.535 | 0.574 | **0.589** | 0.786 | **0.886** | 0.884 | 0.768 | **0.770** | 0.738 | 0.792 | **0.810** | 0.786 |
| *Gridded* | | | | | | | | | | | | |
| ConvCNP (2020) | 0.477 | 0.549 | **0.580** | 0.832 | **0.877** | **0.877** | 0.710 | 0.667 | **0.720** | 0.774 | 0.762 | **0.800** |
| Gridded TNP (2025) | 0.487 | 0.552 | **0.565** | 0.851 | 0.892 | **0.895** | **0.732** | 0.707 | 0.718 | **0.799** | 0.763 | 0.781 |
| *Diffusion / flow* | | | | | | | | | | | | |
| SADI (2025) | 0.565 | **0.604** | 0.552 | 0.756 | 0.822 | **0.880** | 0.519 | 0.607 | **0.665** | 0.485 | 0.532 | **0.768** |
| GiFlow (2026) | 0.438 | **0.510** | 0.466 | 0.800 | 0.826 | **0.860** | **0.699** | 0.652 | 0.688 | **0.770** | 0.731 | 0.744 |
| RP Flow (2026) | 0.462 | 0.550 | **0.553** | 0.737 | 0.820 | **0.823** | 0.647 | 0.664 | **0.666** | 0.645 | 0.726 | **0.727** |
| ZeroDiff (2026) | 0.518 | 0.530 | **0.561** | 0.816 | 0.808 | **0.820** | 0.738 | 0.744 | **0.770** | 0.773 | 0.798 | **0.815** |

Table 5: Kling–Gupta efficiency, station median (higher is better).

| | Streamflow | | | Stream temp. | | | SWE | | | Snow depth | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Model | base | +rep | +RepCal | base | +rep | +RepCal | base | +rep | +RepCal | base | +rep | +RepCal |
| *Neural processes* | | | | | | | | | | | | |
| CNP (2018) | 0.527 | 0.586 | **0.589** | 0.763 | 0.797 | **0.804** | 0.603 | 0.567 | **0.608** | 0.672 | 0.651 | **0.690** |
| TNP (2022) | 0.542 | 0.571 | **0.579** | 0.743 | 0.796 | **0.798** | **0.620** | 0.576 | 0.587 | 0.677 | 0.657 | **0.680** |
| TE-TNP (2024) | 0.530 | 0.570 | **0.605** | 0.736 | 0.803 | **0.805** | 0.613 | 0.611 | **0.620** | 0.622 | **0.632** | 0.617 |
| *Gridded* | | | | | | | | | | | | |
| ConvCNP (2020) | 0.502 | 0.572 | **0.610** | 0.756 | 0.814 | **0.822** | 0.600 | 0.560 | **0.619** | 0.675 | 0.642 | **0.679** |
| Gridded TNP (2025) | 0.495 | 0.532 | **0.560** | 0.763 | **0.826** | 0.814 | **0.618** | 0.578 | 0.587 | **0.669** | 0.633 | 0.639 |
| *Diffusion / flow* | | | | | | | | | | | | |
| SADI (2025) | **0.606** | 0.595 | 0.564 | 0.695 | 0.739 | **0.781** | 0.326 | 0.475 | **0.516** | 0.397 | 0.347 | **0.627** |
| GiFlow (2026) | 0.375 | 0.467 | **0.510** | 0.728 | **0.794** | 0.786 | **0.579** | 0.502 | 0.546 | **0.652** | 0.561 | 0.595 |
| RP Flow (2026) | 0.488 | **0.552** | 0.546 | 0.692 | 0.764 | **0.769** | 0.510 | **0.532** | 0.522 | 0.484 | **0.568** | 0.553 |
| ZeroDiff (2026) | 0.459 | **0.525** | 0.523 | 0.749 | 0.771 | **0.778** | 0.659 | 0.652 | **0.691** | 0.669 | 0.661 | **0.683** |

Table 6: RMSE, station median, in the units of each task (lower is better).

| | Streamflow (mm d$^{-1}$) | | | Stream temp. (°C) | | | SWE (mm) | | | Snow depth (mm) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Model | base | +rep | +RepCal | base | +rep | +RepCal | base | +rep | +RepCal | base | +rep | +RepCal |
| *Neural processes* | | | | | | | | | | | | |
| CNP (2018) | 1.87 | 1.75 | **1.74** | 2.30 | 2.00 | **1.95** | **91.7** | 98.9 | 93.3 | 232.6 | 249.4 | **230.3** |
| TNP (2022) | 1.87 | 1.81 | **1.76** | 2.48 | 2.02 | **1.99** | **92.5** | 100.2 | 92.9 | **233.2** | 235.8 | 233.3 |
| TE-TNP (2024) | 1.80 | 1.75 | **1.70** | 2.48 | 1.90 | **1.88** | 89.4 | **87.7** | 92.6 | 240.3 | **238.0** | 242.6 |
| *Gridded* | | | | | | | | | | | | |
| ConvCNP (2020) | 1.93 | 1.83 | **1.69** | 2.21 | 1.87 | **1.85** | 97.0 | 104.7 | **93.5** | 249.8 | 273.3 | **244.8** |
| Gridded TNP (2025) | 1.90 | **1.76** | 1.82 | 2.10 | **1.89** | **1.89** | **92.0** | 97.0 | 92.6 | **234.2** | 259.7 | 243.9 |
| *Diffusion / flow* | | | | | | | | | | | | |
| SADI (2025) | 1.77 | **1.68** | 1.69 | 2.54 | 2.27 | **1.95** | 118.6 | 109.8 | **104.1** | 379.0 | 354.4 | **264.1** |
| GiFlow (2026) | 1.99 | **1.95** | 2.00 | 2.40 | 2.22 | **2.05** | **100.6** | 112.2 | 107.0 | **261.6** | 287.6 | 288.7 |
| RP Flow (2026) | 1.97 | 1.86 | **1.83** | 2.55 | 2.30 | **2.23** | 111.6 | 110.7 | **109.1** | 331.3 | 294.8 | **291.2** |
| ZeroDiff (2026) | **1.85** | 1.98 | 1.90 | 2.37 | 2.36 | **2.19** | 92.2 | 90.6 | **87.3** | 254.4 | **233.9** | 253.3 |

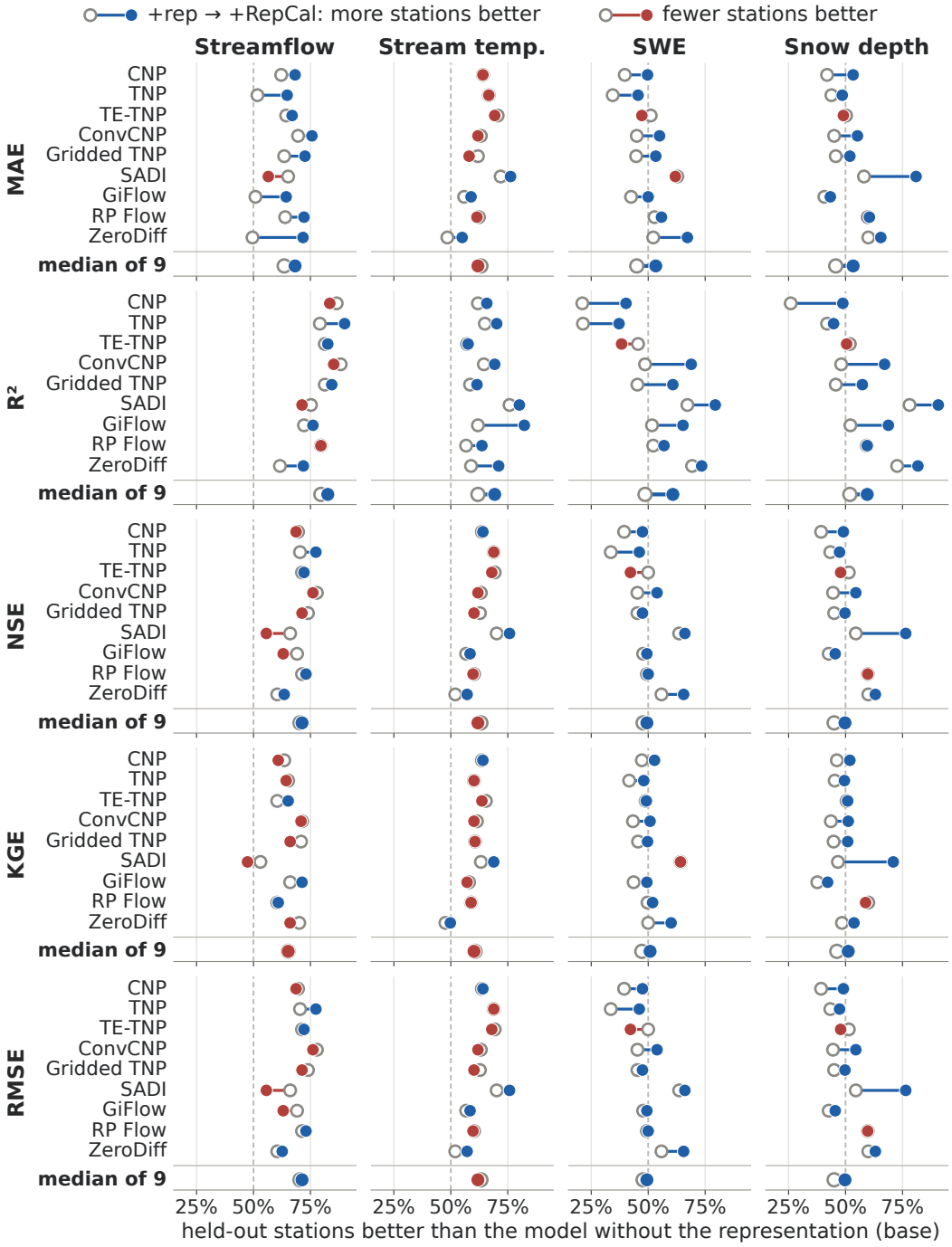


Figure 4: Share of held-out stations of folds 1–7 where each model does better than on its own inputs (base), per task and metric. Open circle: +rep; filled circle: +RepCal. The line between them is blue when +RepCal has more stations better than base, and red when it has fewer. The dashed line marks half of the stations; the last row of each panel is the median over the nine models.

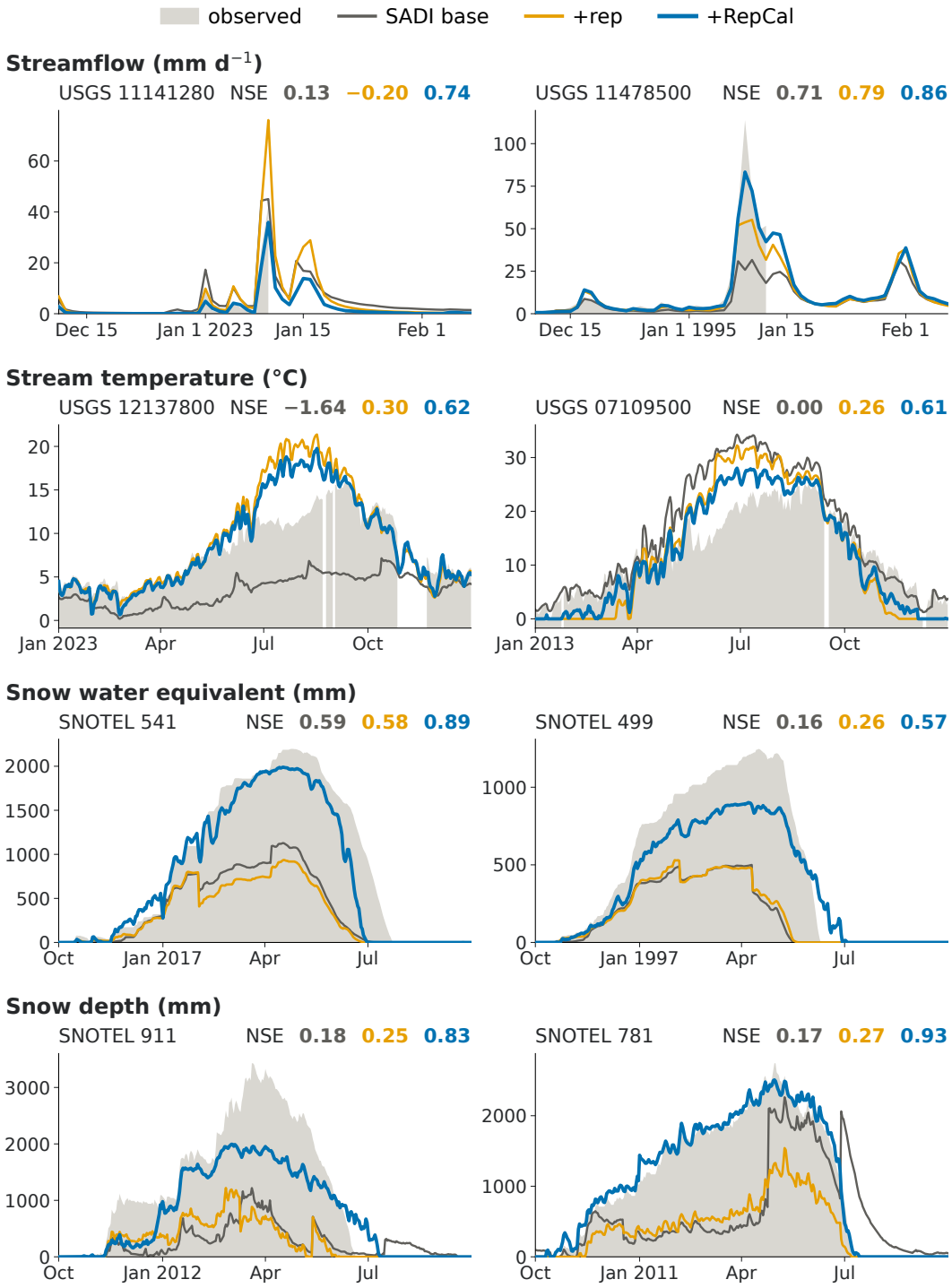


Figure 5: Daily series of SADI at eight held-out stations, two per task. Gray area: observations (white gaps are days without observations); lines: base (its own inputs), +rep (the representation concatenated to its inputs) and +RepCal. The stations were picked among those where +RepCal improves on both base and +rep. Each panel shows the year with the largest variance of the observations, and for streamflow the 60 days around the largest flow of that year. The numbers above each panel are the NSE of the station over its whole record.

# D DATASETS

All sources share one daily time axis, 1985–2024. As in Daymet, every year has 365 days: leap years keep 29 February and drop 31 December.

**Input sources.** BASINGRID (Talha et al., 2026) cuts 9,067 GAGES-II basins into $64 \times 64$ rasters: daily Daymet weather (Thornton et al., 2021), annual potential evapotranspiration and land cover (U.S. Geological Survey, 2024), and static elevation (Gesch et al., 2002), soil texture, organic carbon and bulk density at six depths (Poggio et al., 2021), permeability and hydraulic conductivity (Gleeson et al., 2014; Huscroft et al., 2018), and tile drainage (Valayamkunnath et al., 2020). The encoder uses all layers except land cover and tile drainage (Table 8). CAMELSH (Tran et al., 2025) gives hourly basin-averaged NLDAS-2 forcing (Xia et al., 2012) for 9,008 of these basins, which we average to daily values. The 605 static attributes come from GAGES-II (Falcone, 2011), HydroATLAS (Linke et al., 2019) and NLDAS-2 climate statistics, without those derived from observed streamflow, quality flags and identity codes, to avoid target leakage. The encoder reads all 605, and the downstream models 27 of them.

**Downstream tasks.** Streamflow comes from the USGS gauges in CAMELSH, averaged from hourly to daily values when at least 12 hourly values are present and divided by basin area; stream temperature from USGS gauges (U.S. Geological Survey, 2026); snow water equivalent and snow depth from the SNOTEL network (USDA Natural Resources Conservation Service, National Water and Climate Center, 2026). The candidate stations of a task lie inside a pretraining basin (for streamflow, the 524 CAMELS basins), and we keep those with an observation on more than 85% of the days of the task's time frame (Figure 6b). The kept stations miss 1–7% of their days (Figure 6c); days without an observation contribute no loss.

Table 7: Downstream tasks. The query cell is the area the encoder is asked about for each station (equation 1): the whole basin window for streamflow, which sums the water of the basin upstream of the gauge, and the pixel containing the sensor for the other targets.

| Task | Target (unit) | Query cell | Source | Stations | Years |
|---|---|---|---|---|---|
| Streamflow | discharge ($\mathrm{mm\,d^{-1}}$) | basin window | USGS | 328 | 1992–2024 |
| Stream temperature | water temperature (°C) | sensor pixel | USGS | 329 | 2008–2024 |
| Snow water equivalent | SWE (mm) | sensor pixel | SNOTEL | 513 | 1985–2024 |
| Snow depth | snow depth (mm) | sensor pixel | SNOTEL | 599 | 2005–2024 |

**Inputs.** All four tasks take the same 38 inputs: the 11 basin-averaged NLDAS-2 variables and 27 static basin attributes. No input is a gridded product, and most of the 605 attributes are left out. This is deliberate: the gridded fields and the full attribute table are what the representation summarizes, so every method gets the same inputs and our model sees the rest only through the representation. Snow stations are points inside a basin, often several per basin, and their values depend strongly on elevation. For the two snow tasks we therefore replace the fraction of precipitation falling as snow, a statistic of the target's own climate, with the station's elevation, and predict the moments from the five Daymet weather variables at the station's pixel and the station's elevation (Section 4.2), for every configuration we compare.

**Folds.** The stations of each task are split into ten spatial folds of equal size (Figure 6a), and each fold is held out in turn. Folds 8–10 are the validation set and folds 1–7 the test set (Appendix E.2).

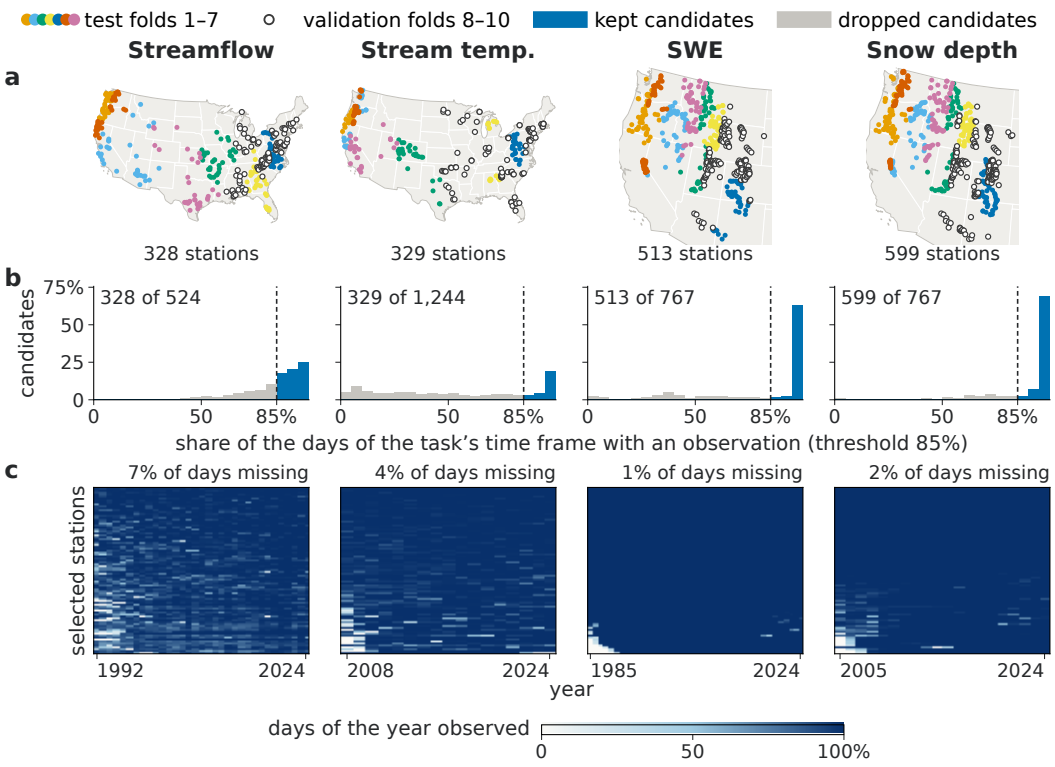


Figure 6: Downstream stations. (a) The kept stations of each task, colored by spatial fold: folds 1–7 are the test set and folds 8–10 (open circles) the validation set. (b) Candidate stations by the share of the days of the task's time frame with an observation; those above 85% (dashed line) are kept. (c) The kept stations (rows, sorted by coverage) by year: the share of the days of each year with an observation.

# E PRETRAINING AND TRAINING DETAILS

This section gives the settings of the encoder (Section 4.1) and of the downstream models (Sections 4.2 and 4.3). Tables 9 and 10 collect the numbers. The downstream settings are the same for all four tasks except the width $d$.

## E.1 PRETRAINING

**Data.** One sample is one basin in one year: the 9,008 basins that appear in BASINGRID, in the NLDAS-2 forcing of CAMELSH and in the attribute table (Appendix D), over 1985–2024. We drop the basin-years that overlap damaged chunks of the BASINGRID store and hold out 2% of the batches to monitor the reconstruction loss. No downstream target is used. The weather series adds variables that the maps lack (pressure, wind, longwave radiation, convective available potential energy, the convective fraction of precipitation and potential evaporation), and where the two overlap, in precipitation and temperature, they come from independent products. Missing attribute values are filled with the column median, and every input is standardized with statistics computed once before training.

Table 8: The four inputs of one pretraining sample and the tokens each becomes.

| Input | Content | Grid | Tokens per sample |
|---|---|---|---|
| Weather maps | 6 Daymet channels: precipitation, maximum and minimum temperature, shortwave radiation, vapor pressure, snow water equivalent | $64 \times 64$, daily | 4,672, one per block of 5 days $\times$ 8 $\times$ 8 pixels |
| Static maps | 34 channels: elevation, permeability, hydraulic conductivity, potential evapotranspiration of the year, and clay, sand, silt, organic carbon and bulk density at six depths | $64 \times 64$ | stacked onto every daily frame, in the same blocks |
| Weather series | 11 NLDAS-2 variables, averaged over the basin | daily | 4,015, one per variable and day |
| Attributes | 605 values from GAGES-II, HydroATLAS and NLDAS-2 climate statistics | one per basin | 8, from learned queries |

**Pixel size.** BASINGRID resamples a padded bounding box of each basin to $64 \times 64$ pixels, so the pixel size follows the basin size over three orders of magnitude, and pixels are stretched (Figure 7a). A pixel stores an average over its area, and nested basins show the same ground at different pixel sizes, so a value at a place and day is only defined together with the area it covers. Every token and query therefore carries its cell size, along $x$ and $y$ separately, as LIIF gives each query the width and height of its pixel (Chen et al., 2021); a single scale, as in scale-aware encoders for square pixels (Reed et al., 2023), cannot describe a stretched pixel.

**Tokens.** Table 8 lists how each input becomes tokens. The map blocks follow the cube embedding of VideoMAE V2 (Wang et al., 2023). A series token is the value times a learned vector of its variable plus a learned embedding of the variable (Nguyen et al., 2023). An attribute is a learned vector scaled by its value plus a learned bias (Gorishniy et al., 2021), and eight learned queries summarize the 605 vectors by cross-attention. Map and series tokens carry their place, day and cell size: the place (NAD83 / CONUS Albers, in kilometers) by sine and cosine features with 16 periods from 8,192 down to 0.25 km, the day by 6 periods from 512 down to 16 days and the first two harmonics of the day of the year, and the cell size by $\log_{10}$ of its width and height in kilometers. A map token sits at the center of its block with one pixel as its cell, and a series token, a basin mean, at the center of the basin window with the whole window as its cell. A learned embedding marks the input each token comes from.

**Encoder and decoder.** $E$ is a pre-norm transformer (Dosovitskiy et al., 2021) with attention over all tokens. $D$ is a stack of cross-attention blocks in which the queries attend to $F_b$ but not to each other (Zhang et al., 2026a; Jaegle et al., 2022). A query is encoded with the same code for place, day and cell size as the tokens, plus a learned embedding of its variable, and a final layer normalization gives $D(q, F_b)$, which $\pi$ maps to a value. The sizes are in Table 9.

**Masking and queries.** A hidden block position stays hidden for the whole year, a tube through time (Wang et al., 2023), and each series variable hides half a year in a row, because a position hidden on some days only, or a short gap in a series, could be filled in from the days around it. The static channels are hidden with their blocks but never predicted, and whole inputs or single series variables are dropped at random. A map query asks for a random pixel of a hidden block, a random day and one of the six Daymet channels, and a series query for a random hidden day. Predicting a sample of the hidden values, not all of them, keeps the decoder cheap, as in the decoder masking of VideoMAE V2.

**Optimization.** A batch holds up to 32 basins of the same year that share a storage chunk of BASINGRID, which makes reading fast. Weight decay applies to all parameters with more than one dimension. The second run continues from the weights of the first with a new optimizer; both warm up linearly and then decay to zero along a cosine (Table 9). Figure 7b shows the held-out loss.

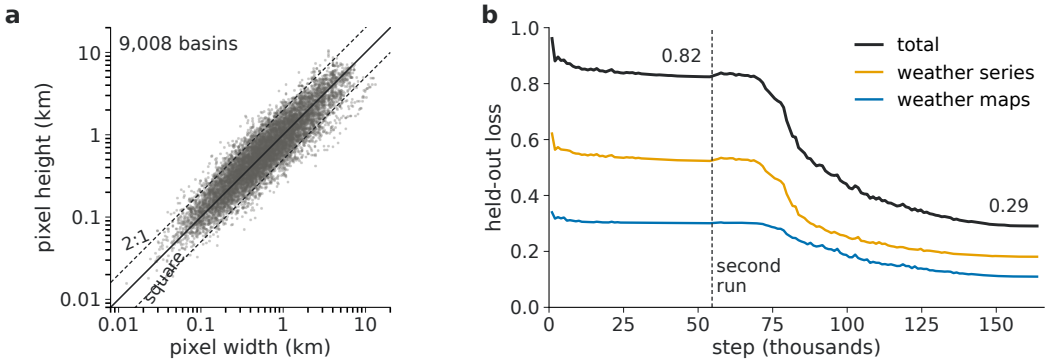


Figure 7: Pretraining. (a) Pixel width and height of the 9,008 pretraining basins; the solid line marks square pixels and the dashed lines a 2:1 ratio. The longer side is 1.33 times the shorter at the median, more than twice as long in 11% of the basins, and at most 5.75 times as long. (b) Held-out reconstruction loss of the two runs, every 1,000 steps: the terms of $\mathcal{L}_{\text{pre}}$ (Section 4.1) for the maps and the series, and their sum. The second run starts from the weights of the first with a new optimizer.

**The representation of a station.** For every basin-year that a station needs, $E$ runs once with every input visible. For each day, we ask one query for each variable in $\mathcal{V}$, the 6 Daymet and 11 NLDAS-2 variables, at the station's place and cell size, and average the 17 features to get $r_{s,t} \in \mathbb{R}^{384}$ of equation 1. One pass of $E$ gives all 365 days for every station in the basin. The downstream targets are never queried.

### E.2 Downstream training

**Samples, folds and validation.** A sample is one station in one year: 365 days of the inputs, the representation and the target, and days without an observation contribute no loss. Because the task is to reconstruct series at stations that are never observed, the training, validation and test sets are split by station, not by time. The stations of each task are divided into ten spatially contiguous folds, and each fold is held out in turn while the model is trained on the other nine, over all years. The held-out stations of folds 8–10 are the validation set: hyperparameters, such as the width $d$ of $W$ from $\{8, 16, 32, d_r\}$, are chosen by the median NSE on these stations, which gives $d = 16$ for stream temperature and $d = d_r$, no projection, for the other three tasks. The held-out stations of folds 1–7 are the test set. They give every reported number and are never used to choose a setting.

**Moments.** The conditional VAE $h_\psi$ is trained on the training stations of each fold and predicts the moments of every station from the mean and standard deviation over time of each input column; for the two snow tasks, of the five Daymet variables at the station's pixel, together with the station's elevation. The predicted moments are the mean over 1,000 draws of the latent from its prior.

**Prior.** In our model, $g_\omega$ is an LSTM with a linear output layer, and $W$ is linear without bias. It is trained on the root mean squared error over observed days of the standardized target for a fixed number of epochs, and the final weights are kept. A batch is one year of every station of the task.

**Calibration.** The regression network $f_\phi$ is the encoder of the non-stationary transformer (Liu et al., 2022). The denoiser embeds the condition, passes it through one-dimensional convolutions and a cross-attention with $y^k$ and $\hat{y}_0$, and then through residual layers conditioned on the step $k$; its two heads give $\epsilon_\theta$ and $\sigma_\theta$, as in NsDiff (Ye et al., 2025). The three terms of equation 17 have equal weights, and stations are weighted as in Appendix B.3. The prior is trained jointly at a learning rate one hundred times smaller; for the +RepCal rows of the baseline experiment it is frozen instead. The weights with the lowest training loss are kept.

**Prediction.** At a held-out station we draw 100 series for each year, each from its own reverse chain started at $y^K = \hat{y}_0 + z$. The point prediction scored by NSE and the other metrics is one chain with $z = 0$ at the start and at every step. One fold of one task, from the prior to the predictions, takes under two hours on one H100 GPU.

Table 9: Pretraining settings.

| Setting | Value |
|---|---|
| *Data* | |
| Basins; years | 9,008; 1985–2024 |
| Basin-years (dropped) | 360,320 (6,353) |
| Held out to monitor the loss | 2% of batches |
| *Tokens* | |
| Map block (days × pixels) | $5 \times 8 \times 8$ |
| Map / series / attribute tokens | 4,672 / 4,015 / 8 |
| *Model* | |
| Encoder layers / heads / width | 12 / 6 / 384 |
| Feed-forward ratio; dropout | 4; 0 |
| Decoder layers / heads | 4 / 6 |
| Parameters of $E$ and $D$ | 34.4 M |
| *Masking* | |
| Hidden block positions | 58 of 64 |
| Hidden days per series variable | 182 in a row |
| Drop rate of each input | 0.1 |
| Map / series queries per sample | 1,024 / 256 |
| *Optimization* | |
| Optimizer $(\beta_1, \beta_2)$ | AdamW (0.9, 0.95) |
| Weight decay; clipping | 0.05; 1.0 |
| Batch | up to 32 basins |
| Precision; hardware | float16; 1 H100 |
| Run 1: steps / warm-up | 54,685 / 5,000 |
| Run 1: peak learning rate | $1.5 \times 10^{-4}$ |
| Run 2: steps / warm-up | 109,370 / 2,000 |
| Run 2: peak learning rate | $10^{-4}$ |
| Decay | cosine to 0 |
| Held-out loss, run 1 / run 2 | 0.82 / 0.29 |
| Compute | 41 H100-hours |

Table 10: Downstream settings.

| Setting | Value |
|---|---|
| *Moments (conditional VAE)* | |
| Hidden / latent width | 128 / 16 |
| Weight of the KL term | 0.1 |
| Optimizer; learning rate | Adam; $10^{-3}$ |
| Batch; epochs | 32; 100 |
| *Prior (LSTM)* | |
| Layers / hidden width | 1 / 20 |
| Input dropout | 0.2 |
| Width $d$: stream temp. / others | 16 / $d_r$ |
| Optimizer; learning rate | Adam; $10^{-3}$ |
| Weight decay; clipping | $10^{-3}$; 3 |
| Batch | one year, all stations |
| Epochs | 150 |
| *Calibration* | |
| $f_\phi$ layers / width / heads | 2 / 512 / 4 |
| $f_\phi$ feed-forward width; dropout | 1,024; 0.05 |
| Denoiser condition width | 32 |
| Denoiser layer widths | 256, 256, 256, 128 |
| Denoiser dropout | 0.1 |
| Diffusion steps $K$ | 20 |
| Schedule of $\beta$ | linear, $10^{-4}$ to $10^{-2}$ |
| Station weights $w_s$; $\tau$ | 0.1 to 2.0; 0.5 |
| Optimizer; precision | Adam; bfloat16 |
| Learning rate: calibration | $10^{-3}$ |
| Learning rate: prior | $10^{-5}$ (0 if frozen) |
| Epochs; patience | at most 200; 20 |
| *Prediction* | |
| Samples per station and year | 100 |
| Point prediction | one chain, $z = 0$ |