# APILANET: ADAPTIVE PHYSICS-INFORMED LATENT NETWORK FOR SINGLE-SENSOR FORECASTING

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

## ABSTRACT

Forecasting conservation-governed dynamics is often constrained by sparse sensing: in practice, we may have only a single boundary sensor and noisy exogenous variables. In this work we design an **A**daptive **P**hysics-**I**nformed **La**tent Network (APILANET) that learns a latent field and enforces 1-D conservation of physics law in the weak form using a learned, normalized space–time measure. Normalization makes physics enforcement insensitive to quadrature resolution and concentrates it on transient violations. A monotone, Lipschitz measurement layer maps latent variables to observed targets, improving identifiability from a single sensor. An adaptive, bounded scheduler scales the physics and smoothness loss terms with meaningful representations, emphasizing conservation of physics laws during events while preserving training stability. Learning a space-time measure for weak-form enforcement, combined with a monotone mapping and adaptive scheduling, enables accurate, data-efficient single-sensor forecasting in physics-governed systems. We evaluate APILANET through a synthetic and hydrological case study, APILANET outperforms strong sequence baselines and reduces MSE during extreme events, while improving Nash–Sutcliffe efficiency. Code will be released upon acceptance.

## 1 INTRODUCTION

Learning the evolution of physical systems from sparse, noisy observations is a central challenge in scientific machine learning. Many natural and engineered processes are governed by partial differential equations (PDEs), yet in practice we often observe only a single location or a few boundary points over time. Examples span climate dynamics Zanella et al. (2023), biomedical flows Ling et al. (2024), battery state-of-health Wang et al. (2025), and river hydraulics. Classical physics-based models typically require dense boundary/interior supervision and careful calibration, while purely data-driven forecasters struggle to extrapolate reliably and to maintain physical consistency over long horizons Nathaniel et al. (2024); Azad et al. (2025).

Physics-Informed Neural Networks (PINNs) Raissi et al. (2019) embed governing laws into learnable models by penalizing PDE residuals. For 1D conservation laws such as

$$\partial_t h(t,x) + \partial_x Q(t,x) \;=\; R_{\text{proj}}(t,x), \tag{1}$$

strong-form PINNs minimize a pointwise residual alongside a data term. This is ill-matched to sparse-observation regimes: (i) it relies on dense interior collocation or full boundary data, (ii) it uses static trade-offs between data and physics losses that can destabilize optimization, and (iii) it offers limited interpretability of learned dynamics and failure modes Kim et al. (2021); Rohrhofer et al. (2023). Recent adaptive weighting schemes (e.g., SA-PINN (McClenny & Braga-Neto, 2023) and ReLoBRaLo (Ling et al., 2024)) rebalance residuals but remain agnostic to real-time signal structure and do not address the lack of spatial supervision.

We propose APILANET, an Adaptive Physics-Informed Latent Neural Network for forecasting PDE-constrained systems from single-point time series. APILANET reconstructs a latent spatiotemporal domain anchored at the observation site and enforces equation 1 in the weak form by integrating residuals against learned test functions rather than penalizing pointwise errors. This lowers regularity requirements, removes the need for interior collocation, and better reflects sensing setups where temporal signals are dense but spatial coverage is sparse.

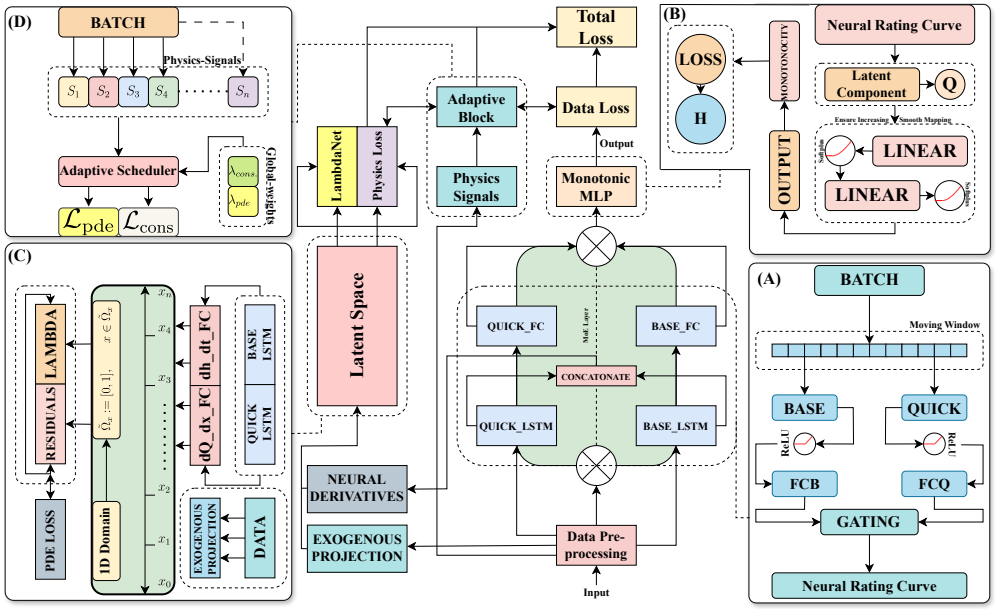



Figure 1: **APILaNet overview.** Single-sensor input window: observed state $h(t)$ and exogenous drivers. A latent 1-D domain $x \in [0,1]$ is instantiated for weak physics. **(A)** Dual streams infer flux components: BASE–LSTM and QUICK–LSTM. A gate $\alpha \in [0,1]$ mixes them, $Q = \alpha Q_{\text{quick}} + (1-\alpha) Q_{\text{base}}$. **(B)** Monotone rating curve $f_{\text{mono}}$ maps mixture of latent components to target $\hat{h} = f_{\text{mono}}(Q)$ with $\partial f_{\text{mono}}/\partial Q \geq 0$ (enforced by a small monotonicity penalty). **(C)** Weak–form physics on the latent mesh: heads predict $\dot{h}_\theta$ and $\partial_x Q_\theta$; a learned weight $\Lambda_\psi(t,x)$ emphasizes where residuals matter. The driver projection $R_\kappa(t,x) = \bar{r}(t)\, e^{-\kappa x}$ injects forcing. Residual $\mathcal{R} = \dot{h}_\theta + \partial_x Q_\theta - R_\kappa$ is penalized in the weak form. **(D)** Adaptive scheduling: bounded signals modulate $\lambda_{\text{pde}}$ and $\lambda_{\text{smooth}}$. Total loss $L = L_{\text{data}} + \lambda_{\text{pde}} L_{\text{pde}} + \lambda_{\text{smooth}} L_{\text{cons}} + \lambda_{\text{mono}} L_{\text{mono}}$.

At a high level, a dual-stream sequence encoder (capturing slow and fast modes) infers a latent conserved flux field $Q_\theta(t,x)$; a monotone neural observation map transforms this latent field into the measured signal at the sensor; and automatic differentiation evaluates the measure-weighted weak-form residual in Eq. equation 2. Training is adaptive: physics penalties are modulated online by bounded signals derived from prediction error, external forcings, and event indicators, increasing conservation pressure during transients and relaxing it in near-stationary regimes. Although our experiments focus on hydrological time series, the architecture is defined at the level of generic 1-D conservation laws under sparse spatial supervision.

$$\mathcal{L}_{\text{PDE}} = \left\| \int_0^1 \left( \partial_t h_\theta(t,x) + \partial_x Q_\theta(t,x) - R_\kappa(t,x) \right) \phi_\psi(t,x)\, dx \right\|_2^2, \tag{2}$$

The contributions of this paper are threefold: (1) APILa framework — a measure-weighted weak formulation for single-sensor learning of 1-D conservation laws on a latent spatial coordinate, instantiated via learned test functions and an equivalent normalized space–time density view, together with a variational dual-stream prior in $H^1$/BV that decomposes slow and fast components of the latent flux; (2) Theory — we provide conditions for single-sensor identifiability under a monotone, Lipschitz observation map and mild excitation of exogenous drivers, prove reparameterization invariance of the weak objective on the latent coordinate, and show the equivalence between the learned-density and learned test-function formulations; (3) Adaptive physics scheduling — a bounded, signal-aware scheme that modulates auxiliary physics terms in time based on task-relevant statistics, tightening conservation during transients and relaxing it in near-stationary regimes. $\lambda_i(t) = \text{clip}\big(\lambda_i^0(1 + \sum_k \alpha_{ik}\, s_k(t)), [\lambda_i^{\min}, \lambda_i^{\max}]\big)$, prioritizing conservation during transients while preserving stability.

We organize the paper as follows: Section 2 reviews related work; Section 3 formalizes the latent weak-form framework and the adaptive training scheme; Section 4 details datasets and protocol; Section 5 concludes.

## 2 Related Work

**Physics-informed learning from sparse observations.** PINNs embed governing laws via residual penalties and have shown wide appeal across scientific domains Raissi et al. (2019). Yet strong-form residuals typically presume dense interior collocation and can be brittle under scarce spatial supervision. Variants that relax regularity or integrate residuals against test functions (weak/variational forms) aim to improve robustness to noise and discretization while reducing collocation burden, but they still require careful loss balancing and often lack guarantees under single-sensor settings (see empirical discussions in Nathaniel et al. (2024); Azad et al. (2025); Rohrhofer et al. (2023)). Training stability in PINNs frequently hinges on the choice of trade-off weights between data and physics losses. Recent adaptive schemes rebalance terms during optimization, e.g., self-adaptive PINNs (SA-PINN) McClenny & Braga-Neto (2023) and ReLoBRaLo Ling et al. (2024), which adjust coefficients based on gradient magnitudes or residual statistics. These methods are largely signal-agnostic and momentum-driven, and they do not exploit domain cues available at run time, such as event likelihood or regime changes, to modulate physics pressure.

For 1-D conservation systems observed at a single site (e.g., stage/discharge), sequence encoders are often used to form latent dynamics, while observation models (rating curves) impose a monotone relationship between discharge and stage. Prior work typically treats the observation link as fixed or unconstrained; monotone neural parameterizations provide a learnable but physically consistent mapping. However, most approaches neither enforce conservation in a weak form over a latent reach nor couple it with adaptive, signal-aware scheduling.

APILANET differs by (i) enforcing a *measure-weighted weak form* on a latent 1-D domain anchored at the observation site, avoiding dense interior collocation; (ii) using a *monotone* learnable rating curve to tie latent discharge to measured stage; and (iii) introducing a *signal-driven* adaptive schedule that modulates auxiliary physics terms online. Together these address sparse spatial supervision, stability, and physical consistency beyond prior PINNs and adaptive-weighting strategies Raissi et al. (2019); McClenny & Braga-Neto (2023); Ling et al. (2024).

### 2.1 Problem setup & notation

Let $\Omega \subset \mathbb{R}^d$ be a bounded Lipschitz domain with horizon $[0,T]$. We model a *latent* state $u : \Omega \times [0,T] \to \mathbb{R}^p$ approximately governed by following equation

$$\partial_t u(x,t) + \nabla \cdot F\big(u(x,t)\big) = S(x,t), \qquad (x,t) \in \Omega \times (0,T), \tag{3}$$

with flux $F : \mathbb{R}^p \to \mathbb{R}^{p\times d}$ and source $S$. Initial/boundary data are $u(\cdot,0) = u_0 \in L^2(\Omega;\mathbb{R}^p)$ and $\mathcal{B}(u, F(u)) = g_{\partial\Omega}$ on $\partial\Omega \times (0,T)$. Exogenous drivers $\xi : [0,T] \to \mathbb{R}^m$ act through a bounded projection

$$S(\cdot,t) = \mathcal{P}_\kappa[\xi](\cdot,t), \qquad \mathcal{P}_\kappa : L^2(0,T;\mathbb{R}^m) \to L^2(\Omega \times (0,T);\mathbb{R}^p), \tag{4}$$

parameterized by $\kappa \in \mathcal{K}$. When $\Omega$ is implicit we work on a latent 1-D chart $(\widehat{\Omega}, \phi)$ with $C^1$ diffeomorphism $\phi : \widehat{\Omega} \to \Omega$; Jacobian factors are absorbed into the sampling/importance measure.

We observe a *single* downstream time series via a bounded linear functional $\mathcal{C} \in (H^1(\Omega;\mathbb{R}^p))^*$ and a shape-constrained measurement map

$$\widehat{y}_\theta(t) = g_\theta\big(\mathcal{C}[u_\theta(\cdot,t)]\big) \in \mathbb{R}, \tag{5}$$

for which we use a monotone, Lipschitz parameterization enforced by architecture. Given observations $y(t_n)$ at $\mathcal{T}_{\text{obs}} = \{t_n\}_{n=1}^N$, the task is: from a history of length $L_{\text{in}}$ and drivers $\xi$, predict $\{y(t_{n+1}), \dots, y(t_{n+L_{\text{out}}})\}$. We write $t_n = n\Delta t$ and $a_{n:n+k} = (a(t_n), \dots, a(t_{n+k}))$; mini-batches are contiguous windows $\big(y_{\,n-L_{\text{in}}:n},\, \xi_{\,n-L_{\text{in}}:n+L_{\text{out}}}\big)$.

For analysis we assume

$$u \in L^2\big(0,T;H^1(\Omega;\mathbb{R}^p)\big) \quad \text{and} \quad \partial_t u \in L^2\big(0,T;H^{-1}(\Omega;\mathbb{R}^p)\big),$$

so the terms in the weak form are well-defined when $F$ is $C^1$ on the range of $u_\theta$. With test functions $\varphi \in H_0^1(\Omega;\mathbb{R}^p)$, multiplying equation 3 by $\varphi$ and integrating by parts in space yields

$$\langle \partial_t u, \varphi\rangle_{H^{-1},H^1} \;-\; \int_\Omega \langle F(u), \nabla\varphi\rangle\, dx \;-\; \int_\Omega S\cdot\varphi\, dx \;=\; 0 \quad \text{for a.e. } t \in (0,T). \tag{6}$$

A *weak solution* of equation 3–$\mathcal{B}$ is $u$ with $u(\cdot,0) = u_0$ satisfying equation 6 for all $\varphi \in H_0^1$ (or for all $\varphi \in H^1$ when nonzero boundary traces are retained), with $S = \mathcal{P}_\kappa[\xi]$. A neural parameterization $u_\theta$ induces $\widehat{y}_\theta$ via equation 5; training penalizes weak-form residuals using a *learned, normalized* space–time importance density $\lambda_\psi : \Omega \times [0,T] \to (0,1]$ with $\iint \lambda_\psi\, dx\, dt = 1$, together with a supervised discrepancy between $y$ and $\widehat{y}_\theta$. The objective (adaptive weights and shape constraints) and training details are given in §A–§D. *Assumptions (compact):* (A1) $F$ is $C^1$ and locally Lipschitz on the range of $u_\theta$; (A2) $\xi \in L^\infty(0,T)$ and $\mathcal{P}_\kappa$ is bounded $L^2 \to L^2$; (A3) $\mathcal{C}$ is bounded and $g_\theta$ satisfies its structural constraint; (A4) $\lambda_\psi \in L^\infty$ and normalized. *Remark.* On graphs, replace $\nabla\cdot$ by $B^\top f$ with incidence matrix $B$; the development is unchanged.

# 3 Method

## 3.1 Panel A: Dual–Stream Latent Dynamics Prior with Input–Driven Gating

From a single–sensor input window $X_{1:L} \in \mathbb{R}^{L\times d}$ we form two *latent flux* sequences over the forecast horizon $\tau = 1{:}T$: a *slow* component $Q_{\text{base}}(\tau)$ and a *fast* component $Q_{\text{quick}}(\tau)$. The encoders that produce these sequences are standard sequence models. We introduce an *input–driven gate* $\alpha \in [0,1]$ and define the latent component passed to sensor location by the convex combination

$$Q_\theta(\tau) \;=\; \alpha\, Q_{\text{quick}}(\tau) \;+\; \big(1-\alpha\big)\, Q_{\text{base}}(\tau), \qquad \alpha = \sigma\big(g(X_{1:L})\big), \tag{7}$$

where $g$ is an arbitrary scalar readout of the history and $\sigma$ is the logistic sigmoid. We enforce $Q_{\text{base}}, Q_{\text{quick}} \geq 0$, hence $Q_\theta \geq 0$ by construction. This single nonnegative $Q_\theta$ is the only latent signal consumed by the observation link and weak physics. To bias the decomposition toward interpretable dynamics, we regularize each component with complementary seminorms:

$$\mathcal{R}_{\text{base}} = \sum_{\tau=2}^{T} \big(\Delta Q_{\text{base}}(\tau)\big)^2, \qquad \mathcal{R}_{\text{quick}} = \sum_{\tau=2}^{T} \big|\Delta Q_{\text{quick}}(\tau)\big|. \tag{8}$$

*Here* $\Delta Q_\cdot(\tau) = Q(\tau) - Q(\tau-1)$. $\mathcal{R}_{\text{base}}$ promotes $\mathrm{H}^1$–type smoothness; $\mathcal{R}_{\text{quick}}$ is a BV/TV prior. These terms are novel in our context as a *paired* Sobolev/BV prior that encourages low–frequency "component" and high–variation "component" within a single latent mixture.

**Assumption 1.** *The history readouts that generate $Q_{\text{base}}, Q_{\text{quick}}$ and the gate $g$ are $L_{\text{b}}, L_{\text{q}}, L_g$–Lipschitz maps w.r.t. $X_{1:L}$.*

**Theorem 1.** *Under A1, for any windows $X, X'$,*

$$\big\|Q_\theta(\cdot;X) - Q_\theta(\cdot;X')\big\|_\infty \;\leq\; \Big(L_{\text{q}}\|\phi_{\text{q}}\| + L_{\text{b}}\|\phi_{\text{b}}\| + \tfrac{1}{4}L_g\,\Delta_Q(X')\Big)\,\|X - X'\|,$$

*where $\Delta_Q(X') = \sup_\tau \big|Q_{\text{quick}}(\tau;X') - Q_{\text{base}}(\tau;X')\big|$. If a uniform bound $\Delta_Q(X') \leq \Delta_{\max}$ holds, replace $\Delta_Q(X')$ by $\Delta_{\max}$.* Proof in Appendix B.

Under mild encoder regularity, the gated mixture $Q_\theta$ in equation 7 is Lipschitz in the input window, so small changes in $X_{1:L}$ yield bounded changes in the latent component. Moreover, the paired Sobolev/BV priors in equation 8 induce a Tikhonov–TV splitting that assigns low-frequency content to $Q_{\text{base}}$ and high-variation content to $Q_{\text{quick}}$. Formal statements and proofs are provided in (Appendix B).

## 3.2 Panel B: Monotone Latent Mapping

Panel B maps the aggregated *driver* from Panel A to the observed *target* using a shallow neural link *without* assuming any fixed parametric law. Concretely, a bias-enabled two-layer MLP with Softplus activations is applied element-wise in time to the clamped (nonnegative) driver. The biases absorb sensor offsets and the flexible link avoids imposing a fixed power-law shape. We

introduce (i) an *empirical, order-preserving monotonicity surrogate* that enforces a nondecreasing driver-to-target map on the *observed* driver range without constraining weights, and (ii) a *consistency* statement showing that, as design points densify, vanishing surrogate loss yields almost-everywhere monotonicity over the training range.

Given a finite set $\mathbf{q} = \{q_i\}_{i=1}^{n}$ from the (clamped) driver range with $q_{(1)} \le \cdots \le q_{(n)}$, define $\mathcal{L}_{\text{mono}}(\theta; \mathbf{q}) = \dfrac{1}{n-1} \sum_{i=1}^{n-1} \big[f_\theta(q_{(i+1)}) - f_\theta(q_{(i)})\big]_-$, with $[x]_- = \max\{0, -x\}$. We add $\gamma_{\text{mono}} \mathcal{L}_{\text{mono}}$ to the loss ($\gamma_{\text{mono}}$=0.01).

**Proposition 1.** *$\mathcal{L}_{\text{mono}}(\theta; \mathbf{q}) = 0$ if $f_\theta(q_{(i+1)}) \ge f_\theta(q_{(i)})$ for all adjacent pairs. Moreover, $\max_i [f_\theta(q_{(i)}) - f_\theta(q_{(i+1)})]_+ \le (n-1)\, \mathcal{L}_{\text{mono}}(\theta; \mathbf{q})$.*

If design sets $\mathbf{q}^{(m)} \subset [0, Q_{\max}]$ densify, $\sup_m \|f_{\theta_m}\|_\infty < \infty$, and a standard regularizer yields a uniform total-variation bound, then a subsequence converges pointwise a.e. to a monotone limit on $[0, Q_{\max}]$ when $\mathcal{L}_{\text{mono}}(\theta_m; \mathbf{q}^{(m)}) \to 0$. Together, this surrogate-and-proof package gives a lightweight way to impose a domain-plausible monotone observation link *only where the data live*, improving identifiability and training stability without hard weight constraints.

### 3.3 Panel C: Weak-Form Physics on the Latent Mesh

We enforce a conservation law in a *latent* spatiotemporal domain using only single-point time series. Concretely, the model predicts two time-indexed sequences, an objective-time derivative $d_t h_\theta[\tau]$ and an exogenous-space derivative $d_x Q_\theta[\tau]$ and broadcasts them across a fixed $X$-cell latent spatial grid. The exogenous variable is projected over this grid via a learnable, monotone spatial kernel. The weak-form loss is the average of squared residuals weighted by a learned, non-negative field. We introduce (i) a *broadcast weak-form* residual on a latent mesh that turns single-point supervision into spatiotemporal physics via broadcasting and exogenous-variable projection; (ii) an *exponential exogenous projection* with learnable decay $\kappa > 0$ enabling spatial structure from a point variable; (iii) a *learned spatial weighting field* that emphasizes informative cells while remaining non-negative by construction.

**From classical weak form to APILaNet's latent weak form.** We compare (i) the classical weak residual with constant test functions on a 1D strip, and (ii) our broadcast residual on a latent mesh with a learned, normalized weight.

**Assumption 2** (Proxy derivatives and latent forcing)**.** *For each forecast step $\tau \in \{1{:}T\}$, the model outputs proxies $d_t h_\theta[\tau] \approx \partial_t h(\tau, \cdot)$ and $d_x Q_\theta[\tau] \approx \partial_x Q(\tau, \cdot)$ that are (piecewise) constant in $x$ when broadcast across a latent grid $\{x_j\}_{j=1}^{X} \subset [0,1]$. A single exogenous series is projected to a latent forcing $R_\theta(x) = \bar{R}\, \mathrm{e}^{-\kappa x}$ with $\kappa > 0$ learnable.*

**Assumption 3** (Learned, normalized measure)**.** *A nonnegative field $\lambda_\phi(x) \ge 0$ induces a measure $\mathrm{d}\mu_\phi(x) = \lambda_\phi(x)\, \mathrm{d}x$ on $[0,1]$ that is (i) bounded and bounded away from 0 on compact subsets, and (ii) normalized so that $\int_0^1 \lambda_\phi(x)\, \mathrm{d}x = 1$.*

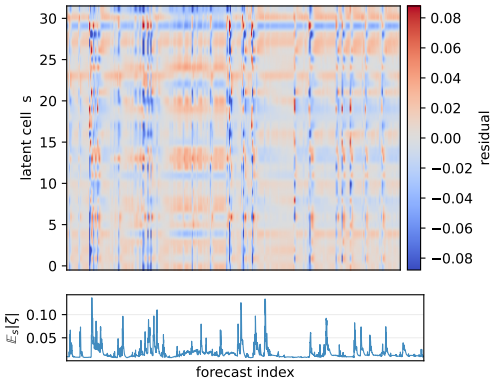



Figure 2: Weak–form residual heat map $\zeta(t, s)$ with per-step mean $\mathbb{E}_s|\zeta|$.

Figure 2 visualizes the weak–form residual $\zeta(t, s) = \partial_t h + \partial_x Q - R$ over the latent mesh. Hot/cold bands in the heat map mark where conservation is violated in time ($t$) and across latent cells ($s$); sharp vertical streaks coincide with rapid changes in the driving signal, showing that APILaNet localizes transient imbalance rather than spreading it uniformly. The bottom trace aggregates $\mathbb{E}_s[\,|\zeta|\,]$ and highlights when violations spike, which typically precedes or aligns with observed extremes. This diagnostic is useful both for model debugging, to identify how residuals concentrate during

rare, high-amplitude regimes, and for interpretability (how the model "spends" its physics budget over the prediction horizon).

**Theorem 2** (Reduction to classical weak form)**.** *Under Assumptions 2–3, the APILaNet broadcast loss*

$$\mathcal{L}_{\text{pde}}(\theta,\phi) = \frac{1}{TX}\sum_{\tau=1}^{T}\sum_{j=1}^{X}\lambda_\phi(x_j)\left(d_t h_\theta[\tau] + d_x Q_\theta[\tau] - R_\theta(x_j)\right)^2$$

*is a Riemann (cell-wise) quadrature of the classical weak $L^2(\mu_\phi)$ residual of the continuity law with constant test functions on each cell. In particular, as the latent grid refines ($\max_j |x_{j+1}-x_j| \to 0$),*

$$\mathcal{L}_{\text{pde}}(\theta,\phi) \;\to\; \frac{1}{T}\sum_{\tau=1}^{T}\int_0^1 \left(\partial_t h_\theta(\tau,x) + \partial_x Q_\theta(\tau,x) - R_\theta(x)\right)^2 \mathrm{d}\mu_\phi(x).$$

Proof sketch. *Broadcasting makes the trial/test functions piecewise constant in $x$; averaging over $j$ with weights $\lambda_\phi(x_j)$ is a normalized quadrature for the weighted $L^2$ norm.*

**Adaptive weighting map.** Figure 3 visualizes the learned space–time weight $\lambda(t,s)$ used in the weak-form loss. The heat map shows that $\lambda$ is not uniform: it concentrates near informative regions of the forecast (earlier prediction steps and selected latent spatial cells) and decays elsewhere, indicating that the model allocates more penalty to transient, high-signal zones. The bottom marginal $\mathbb{E}_s[\lambda](t)$ summarizes this temporal emphasis, typically highest near the start of the horizon and tapering with $t$, while the right marginal $\mathbb{E}_t[\lambda](s)$ captures how weighting varies across the latent spatial index. Together with Fig. 2, this confirms that APILaNet *both* locates residual spikes and adaptively "spends" its physics budget where it matters most.

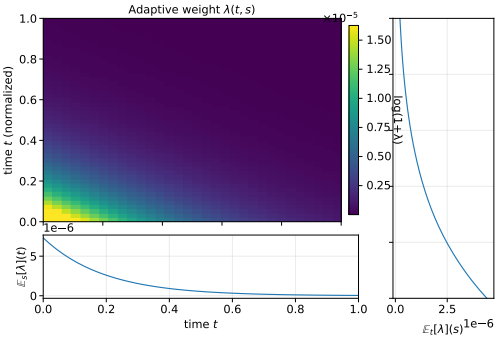



Figure 3: Adaptive weight field $\lambda(t,s)$ learned for the weak form. Left: heat map over time $t$ and latent cell $s$. Bottom: temporal marginal $\mathbb{E}_s[\lambda](t)$. Right: spatial marginal $\mathbb{E}_t[\lambda](s)$. The field assigns larger weight where the dynamics change rapidly and smaller weight in nearly stationary periods.

**Interpretation.** Theorem 2 says our broadcast loss is not an ad-hoc penalty: it is exactly a cell-wise quadrature of the classical weak residual under a learned, normalized measure. In plain terms, APILaNet turns a single-sensor sequence into a principled weak-form discretization on a latent mesh, while $\lambda_\phi$ acts as an importance map that concentrates physics where the signal is informative. Refinement/consistency assumptions and results—namely Assumption 4 (approximation and mesh refinement), Theorem 3 (consistency under refinement), and Corollary 1 (single-sensor realizability through the monotone observation link)—are stated and proved in Appendix D.

### 3.4 Panel D: Adaptive Physics Scheduling

Panel D modulates physics strength. Two global multipliers act on the physics terms: a PDE weight $\lambda_{\text{pde}}$ and a derivative-consistency weight $\lambda_{\text{cons}}$. Each is computed *instantaneously per minibatch* from available signals. In addition, a *local* nonnegative field $\lambda_{\text{loc}}(t,x)$ weights the PDE residual over the latent mesh (Panel C). The effective PDE weight is $\Lambda_{\text{pde}}(t,x) = \lambda_{\text{pde}}\,\lambda_{\text{loc}}(t,x)$. **Objective:** allocate physics pressure *when* and *where* it matters without destabilizing training. We therefore factorize the PDE weight into a *global* batch scalar and a *local* nonnegative field over the latent mesh:

$$\Lambda_{\text{pde}}(t,x) = \lambda_{\text{pde}}\,\lambda_{\text{loc}}(t,x), \quad \lambda_{\text{loc}}(t,x) \geq 0, \quad \frac{1}{TX}\sum_{\tau=1}^{T}\sum_{j=1}^{X}\lambda_{\text{loc}}(\tau,x_j) = 1. \tag{9}$$

The effective PDE term in the loss is

$$\mathcal{L}^{\text{eff}}_{\text{pde}} = \lambda_{\text{pde}}\cdot\frac{1}{TX}\sum_{\tau=1}^{T}\sum_{j=1}^{X}\lambda_{\text{loc}}(\tau,x_j)\,r_\theta[\tau,j]^2, \quad r_\theta[\tau,j] = \partial_t h_\theta[\tau] + \partial_x Q_\theta[\tau] - R_\theta(x_j). \tag{10}$$

**Algorithm 1:** Adaptive Multi-Loss Scheduling with Factorized Local Weights

**Inputs:** mini-batch $\mathcal{D}$, model $\mathcal{F}_\theta$, optimizer; bases $\{\lambda_i^0\}$; sensitivities $\{\alpha_{ik}\}$; clips $[\lambda_i^{\min}, \lambda_i^{\max}]$
**Outputs:** updated parameters $\theta$
**for** *epoch* $e = 1$ **to** $N_{epoch}$ **do**
  **foreach** *mini-batch* $\mathcal{D}$ **do**
    compute per-losses $\{\mathcal{L}_i(\theta, \mathcal{D})\}_{i=1}^m$; optional local map $W_{\text{loc}} \geq 0$
    compute batch signals $\{s_k(\mathcal{D})\}_{k=1}^K$ and activity $\Pi$
    **for** $i = 1$ **to** $m$ **do**
      $\lambda_i \leftarrow \text{clip}\Big(\lambda_i^0\big(1 + \sum_{k=1}^K \alpha_{ik}\, s_k + \alpha_{i,\Pi}\Pi\big), \lambda_i^{\min}, \lambda_i^{\max}\Big)$
    **if** $W_{loc}$ *used* **then**
      $Z \leftarrow \frac{1}{|\Omega|}\sum_{(t,x)\in\Omega} W_{\text{loc}}(t,x)$;
      $W_{\text{loc}} \leftarrow W_{\text{loc}}/Z$
    $\mathcal{L}_{\text{tot}} \leftarrow \sum_{i=1}^m \lambda_i\, \mathcal{L}_i(\theta, \mathcal{D}; W_{\text{loc}})$
    optimizer.zero_grad();
    backprop($\mathcal{L}_{\text{tot}}$);
    optimizer.step()

**Instantaneous global scheduler.** Let $E \geq 0$ be the batch prediction loss, $\mathbf{s} \in \mathbb{R}^K_{\geq 0}$ a vector of auxiliary regime signals, and $\Pi \in [0,1]$ an activity score. For $i \in \{\text{pde}, \text{cons}\}$ we set

$$\lambda_i = \text{clip}\Big(\lambda_i^0\,\big(1 + E + \boldsymbol{\alpha}_i^\top \mathbf{s} + \alpha_{i,\Pi}\,\Pi\big),\ \lambda_i^{\min},\ \lambda_i^{\max}\Big), \tag{11}$$

where $\lambda_i^0 > 0$ is a base level, $(\boldsymbol{\alpha}_i, \alpha_{i,\Pi}) \geq 0$ are sensitivities, and clip enforces user-specified bounds. In the implementation we use this update rule: for each mini-batch we compute $(E, \mathbf{s}, \Pi)$ from the current data, plug them into equation 11, and recompute $\lambda_i$ from scratch.

**Assumption 4** (Bounded signals & normalized local field)**.** *During training, $E$, each component of $\mathbf{s}$, and $\Pi$ are bounded; the local field satisfies equation 9; and equation 11 produces $\lambda_i \in [\lambda_i^{\min}, \lambda_i^{\max}]$.*

**Theorem 3** (Monotone responsiveness with bounded pressure)**.** *Under Assumption 6, each $\lambda_i$ in equation 11 is nondecreasing in $E$, every component of $\mathbf{s}$, and $\Pi$ (away from clips) and always satisfies $\lambda_i^{\min} \leq \lambda_i \leq \lambda_i^{\max}$. Consequently equation 10 is both* responsive *to harder-regime batches and* bounded *to avoid instability.*

**Implementation and hyperparameters.** For clarity, we make the full set of scheduler scalars explicit. For each loss $i \in \{\text{pde}, \text{cons}\}$ we specify base levels $\lambda_i^0$, clipping bounds $(\lambda_i^{\min}, \lambda_i^{\max})$, and nonnegative sensitivities $(\boldsymbol{\alpha}_i, \alpha_{i,\Pi})$. All values used in our experiments are listed in Appendix W2. The only scalars selected by validation are a global physics scale $\lambda_{\text{scale}}$ that multiplies $(\lambda_{\text{pde}}^0, \lambda_{\text{cons}}^0)$ and an activity sensitivity $\alpha_\Pi$ applied to $\Pi$; we choose $(\lambda_{\text{scale}}, \alpha_\Pi)$ once by a small grid search on the validation NSE and then reuse the same pair for all datasets in the corresponding benchmark. All other modulation is purely data–driven through $(E, \mathbf{s}, \Pi)$.

**Sensitivity and robustness.** To assess robustness, we perform a scheduler ablation on a synthetic single-sensor benchmark (Appendix D2), varying $\lambda_{\text{scale}} \in \{0.5, 1.0, 2.0\}$ and $\alpha_\Pi \in \{0, 0.3, 0.6\}$ and comparing adaptive ($\boldsymbol{\alpha}_i > 0$) versus static ($\boldsymbol{\alpha}_i$=0) weights. Across this grid, test MSE and NSE vary smoothly, with no training collapse, and the performance differences between adaptive and static global weights are modest. This indicates that the scheduler does not rely on finely tuned coefficients; its main effect is to redistribute physics pressure towards difficult regimes rather than to optimise aggregate error. Full numerical results are reported in Table D2.

We scale physics by two knobs: a *global*, batch-wise multiplier that grows when the batch looks hard (big errors, event cues) but remains clipped, and a *local*, nonnegative map over the latent mesh that redistributes this budget to where residuals matter. The global rule makes physics *responsive* yet *bounded*; the local normalization preserves the average strength while focusing effort in time–space.

Theorem 7 formalizes this: the scheduler is monotone in difficulty signals away from clips, and the weights stay within $[\lambda_i^{\min}, \lambda_i^{\max}]$, so training remains stable even during sharp transients.

# 4 EXPERIMENTS

## 4.1 PROTOCOLS

**Datasets** We conduct a hydrology case study and experiments on six real–world, single–sensor benchmarks from UK catchments. We construct the same $L \times d$ input tensor for all sites using a unified pipeline. The train/val/test configuration splits for each dataset are same. Addtionaly, we include a general 1D PDE benchmarks (viscous Burgers, wave, Allen–Cahn), where high–resolution reference solutions are generated with a finite–difference solver.

**Baselines** We benchmark APILANET against eight competitive sequence-to-sequence forecasters that span the main families of modern time–series modeling: Transformer Utilizing Cross-Dimension Dependency for Multivariate Time Series Forecasting *CrossFormer* Zhang & Yan (2023); patchwise Transformer *PatchTST* Nie et al. (2023); MLP token–mixer *TS-Mixer* Chen et al. (2023); convolutional token–mixer *PatchMixer* Gong et al. (2023); selective state–space model *Mamba-S4* Dao & Gu (2024); *iTransformer* Liu et al. (2023); and the neural decomposition methods *N-HiTS* Challu et al. (2022) and *N-BEATS* Oreshkin et al. (2020).

**Setup.** All models ingest the same $L \times d$ input tensor and predict the same $T$-step horizon. Inputs are feature-wise *min–max scaled* using statistics computed on the training split and applied to val/test. We generate input–output pairs with a sliding window. We evaluate a fixed forecast horizon $T{=}32$ and look-back length $L{=}32$ based on Table 2. Primary metrics are Mean Squared Error (MSE) and Nash–Sutcliffe Efficiency (NSE); for event-focused analyses we additionally report peak-timing and peak-magnitude errors $(\Delta t_{\text{peak}}, \Delta h_{\text{peak}})$. Baselines use the *same* inputs as APILANET and follow the original authors' recommended model sizes, optimizers, and regularization. All methods are trained for the same epochs, batch size, and learning-rate schedule. Each configuration is run with *three fixed random seeds*; and the mean of the metrics is reported. Full dataset details, implementation, and hyperparameters appear in Appendix A.

## 4.2 ABLATION STUDY

**Ablation Design** We report seven variants corresponding to Table 1: (1) **APILaNet** (full model); (2) *w/o λ Adapt. (global)*; (3) *w/o $\lambda_g$ Adapt. (local)*—remove the *local* weighting (set $\lambda_g{\equiv}1$) while keeping the global scheduler $\lambda_s$ and the PDE loss; (4) *w/o $\lambda_s$ Adapt. (both)*—freeze both weights (fix $\lambda_g{=}\lambda_g^0$ and $\lambda_g{\equiv}1$) with the PDE loss retained; (5) *w/o Monotone MLP*—replace the monotone rating-curve link by an unconstrained scalar MLP; (6) *w/o PDE loss*—drop the weak-form continuity residual from the objective; (7) $\mathcal{L}_{data}$ *only*—pure data fit.

Table 1: Ablation at 8 h before extreme event on Stocksfield. Entries are *mean±SD **[95% CI]*** across seeds. MSE is reported in $\times 10^{-1}$. Best results are **red**; second-best are **blue**.

| Model | $\lambda_g$ | $\lambda_s$ | PDE | $\Delta t_{\text{peak}}$ (h)↓ | $\Delta h_{\text{peak}}$ (m)↓ | MSE ($\times 10^{-1}$) ↓ | NSE↑ |
|---|---|---|---|---|---|---|---|
| (1) **APILANET** | ✓ | ✓ | ✓ | **0.00±0.00** [0.00, 0.00] | 0.46±0.19 [0.18, 0.75] | **0.45±0.14** [0.25, 0.65] | **0.51±0.15** [0.29, 0.72] |
| (2) w/o λ Adapt. (a) | × | × | ✓ | **0.00±0.00** [0.00, 0.00] | 0.46±0.08 [0.33, 0.59] | **0.53±0.06** [0.45, 0.62] | **0.42±0.06** [0.33, 0.51] |
| (3) w/o λ Adapt. (b) | × | ✓ | ✓ | **0.00±0.00** [0.00, 0.00] | **0.39±0.17** [0.13, 0.64] | 0.57±0.03 [0.52, 0.61] | 0.38±0.03 [0.33, 0.43] |
| (4) w/o λ Adapt. (c) | ✓ | × | ✓ | **0.00±0.00** [0.00, 0.00] | 0.52±0.07 [0.41, 0.63] | 0.55±0.07 [0.45, 0.65] | 0.39±0.07 [0.29, 0.50] |
| (5) w/o Mono MLP | ✓ | ✓ | ✓ | **0.00±0.00** [0.00, 0.00] | 0.51±0.16 [0.27, 0.75] | **0.53±0.04** [0.47, 0.59] | 0.41±0.04 [0.35, 0.48] |
| (6) w/o PDE Loss | ✓ | ✓ | × | **0.25±0.42** [-0.19, 0.69] | **0.40±0.14** [0.25, 0.54] | 0.64±0.27 [0.36, 0.93] | 0.29±0.29 [-0.01, 0.61] |
| (7) APILANET $\mathcal{L}_{\text{data}}$ | × | × | × | 1.92±3.32 [-3.01, 6.84] | 0.68±0.24 [0.32, 1.04] | 0.74±0.35 [0.22, 1.26] | 0.19±0.38 [-0.37, 0.76] |

Based on the results from Table 1 , the full APILANET achieves the best MSE/NSE. Removing adaptive weighting degrades accuracy—*both* schedulers matter: using only the $\lambda_g$ or only the $\lambda_s$ field is inferior to using them together. Eliminating the PDE weak–form loss yields the largest drop in peak timing and overall fit, while removing the monotone link also hurts MSE/NSE and stability. Overall, gains are *additive*: monotone link + PDE loss + $(\lambda_g \oplus \lambda_s)$ scheduling produce the strongest performance.

**Sensitivity to latent mesh size and learned measure.** We additionally vary the number of latent cells $X \in \{8, 16, 32, 64\}$ and compare (i) a uniform measure $\lambda_{\text{uni}}(t, x)$ and (ii) the learned measure $\lambda_\phi(t, x)$ (Table 6; full results in App. F). The uniform baseline aggregates performance across all $X$ with a fixed, non–adaptive measure, while the learned $\lambda_\phi$ is trained separately for each resolution $X$. Across all tested resolutions, the learned measure *never underperforms* the uniform baseline: the largest gains occur at moderate resolutions ($X = 16, 32$), with test MSE reduced by roughly 15–18% and NSE improved by about 0.015–0.017. For coarser or finer grids ($X = 8$ or 64), the gains are smaller but remain non–negative.

### 4.3 Influence of Input Sequence Length

Table 2 shows that a medium context is consistently best. Across all five catchments, the optimal lookback is *32 steps* (8 h at 15 min resolution): it yields the lowest MSE and the highest NSE in every case (Acomb Mfs $0.021\times10^{-2}$ / 0.936, Stocksfield $0.053\times10^{-2}$ / 0.886). Short histories ($\leq$16 steps) underfit transients and hurt NSE, while very long histories ($\geq$128) plateau or slightly degrade, likely due to memory dilution, heavier optimization, and fewer distinct windows per epoch. The result is robust—64–128 steps are typically within a few percent of the best—but 32 steps offers the best accuracy–efficiency trade-off. *We therefore fix the lookback to 32 steps (8 h) in all remaining experiments unless stated otherwise.*

Table 2: Lookback sensitivity by catchment. Mean MSE ($\downarrow$, $\times10^{-2}$) and NSE ($\uparrow$) across seven input horizons (2–128 h).

| Site | Metric | Lookback window (time steps) | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | 8 | 16 | 32 | 64 | 128 | 256 | 512 |
| Acomb Grn | MSE ($\times10^{-2}$) | 0.066 | 0.059 | **0.041** | 0.043 | 0.045 | 0.057 | **0.042** |
| | NSE | 0.857 | 0.873 | **0.911** | 0.906 | 0.909 | 0.895 | **0.910** |
| Acomb Mfs | MSE ($\times10^{-2}$) | 0.049 | 0.037 | **0.021** | 0.023 | 0.027 | **0.022** | 0.028 |
| | NSE | 0.853 | 0.888 | **0.936** | 0.931 | 0.919 | **0.933** | 0.916 |
| Stocksfield | MSE ($\times10^{-2}$) | 0.079 | 0.071 | **0.053** | 0.069 | 0.068 | 0.068 | **0.061** |
| | NSE | 0.837 | 0.849 | **0.886** | 0.852 | 0.856 | 0.855 | **0.872** |
| Nunnykirk | MSE ($\times10^{-2}$) | 0.091 | **0.073** | **0.067** | 0.073 | 0.086 | 0.090 | 0.091 |
| | NSE | 0.913 | **0.941** | **0.959** | 0.940 | 0.921 | 0.914 | 0.913 |
| Knitsley | MSE ($\times10^{-2}$) | 0.063 | 0.038 | **0.030** | 0.038 | **0.033** | 0.064 | 0.072 |
| | NSE | 0.915 | 0.936 | **0.946** | 0.935 | **0.943** | 0.912 | 0.902 |
| Kielder | MSE ($\times10^{-2}$) | 0.076 | 0.066 | **0.030** | **0.041** | 0.042 | 0.067 | 0.073 |
| | NSE | 0.898 | 0.912 | **0.962** | **0.944** | 0.943 | 0.908 | 0.902 |

### 4.4 Synthetic 1D PDE Benchmarks

To test whether APILaNet is tied to a single application domain, we also evaluate it on three well-known 1D PDEs: viscous Burgers, the wave equation, and Allen–Cahn. For each, we generate a finite-difference reference solution with standard IC/BC and train vanilla PINN, PINN-w, gPINN, and vPINN in the usual setting with full geometry and interior collocation points, while APILaNet only observes a single probe time series and known forcing, enforcing the conservation law on a latent spatial coordinate (Sec. 3.3). Table 3 reports test MSE at the probe; across all three PDEs, APILaNet matches or outperforms these strong-form and adaptive PINNs despite the weaker information regime, supporting its role as a general single-sensor conservation-law framework.

Table 3: Synthetic 1D PDE benchmarks. Entries are test MSE (lower is better). Best results are **red**; second-best are **blue**.

| PDE (MSE) | Vanilla PINN | PINN-w Ryck et al. (2022) | gPINN Yu et al. (2022) | vPINN Kharazmi et al. (2019) | APILaNet |
|---|---|---|---|---|---|
| Burgers | $5.80 \times 10^{-4}$ | $2.91 \times 10^{-3}$ | $1.29 \times 10^{-4}$ | $1.45 \times 10^{-3}$ | $4.50 \times 10^{-5}$ |
| Wave | $2.62 \times 10^{-4}$ | $2.89 \times 10^{-3}$ | $1.62 \times 10^{-4}$ | $8.91 \times 10^{-4}$ | $1.52 \times 10^{-4}$ |
| Allen–Cahn | $1.18 \times 10^{0}$ | $1.04 \times 10^{0}$ | $1.32 \times 10^{-1}$ | $1.04 \times 10^{0}$ | $1.18 \times 10^{-1}$ |

### 4.5 Additional Experiments

Beyond standard test-set accuracy, we benchmark *early-warning* performance by evaluating every model's ability to predict before the extreme event. This stress test probes how well a forecaster anticipates extremes as lead time shortens—crucial for actionable response. Across all lead times, APILaNet delivers the lowest MSE and highest NSE in most catchments, while also minimizing peak *timing* and *magnitude* errors ($\Delta t_{\text{peak}}$, $\Delta h_{\text{peak}}$). Notably, performance degrades *gracefully* as the warning window widens (8 h $\rightarrow$ 2 h), indicating stable physics-aware generalization rather than last-minute correction. These results suggest APILaNet provides earlier and more reliable alerts than state-of-the-arts baselines, making it better aligned with real-world decision timelines for real-world preparedness and incident management. (Appendix F).

Table 4: Catchment-level forecasting. Test-set MSE (↓) and NSE (↑) across six UK catchments and three events per catchment, with fixed prediction length and horizon.Best results are **red**; second-best are **blue**.

| Data | Model | APILANET | | CROSSFORMER | | PATCHTST | | TSMIXER | | PATCHMIXER | | MAMBA S4 | | ITRANSFORMER | | N-HITS | | N-BEATS | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Metrics | MSE↓ | NSE↑ | MSE↓ | NSE↑ | MSE↓ | NSE↑ | MSE↓ | NSE↑ | MSE↓ | NSE↑ | MSE↓ | NSE↑ | MSE↓ | NSE↑ | MSE↓ | NSE↑ | MSE↓ | NSE↑ |
| ACOMB GRN | Event 1 | **0.090** | **0.810** | **0.117** | **0.754** | 0.471 | 0.009 | 0.127 | 0.733 | 0.117 | 0.753 | 0.317 | 0.333 | 0.122 | 0.744 | 0.362 | 0.238 | 0.337 | 0.290 |
| | Event 2 | **0.058** | **0.919** | 0.093 | 0.869 | 0.385 | 0.460 | **0.073** | **0.897** | 0.082 | 0.884 | 0.222 | 0.689 | 0.106 | 0.851 | 0.341 | 0.522 | 0.311 | 0.564 |
| | Event 3 | **0.935** | **0.329** | 0.951 | 0.318 | 2.485 | -0.783 | **0.926** | **0.335** | 1.514 | -0.087 | 1.357 | 0.026 | 0.968 | 0.305 | 1.682 | -0.207 | 1.712 | -0.229 |
| | **Test** | **0.010** | **0.907** | 0.011 | 0.901 | 0.026 | 0.762 | **0.010** | **0.904** | 0.013 | 0.876 | 0.016 | 0.852 | 0.011 | 0.897 | 0.020 | 0.815 | 0.019 | 0.821 |
| ACOMB MFS | Event 1 | **0.054** | **0.885** | 0.077 | 0.836 | 0.443 | 0.061 | **0.052** | **0.890** | 0.064 | 0.863 | 0.324 | 0.314 | 0.103 | 0.781 | 0.382 | 0.191 | 0.428 | 0.092 |
| | Event 2 | **0.018** | **0.970** | 0.058 | 0.902 | 0.326 | 0.450 | **0.025** | **0.957** | 0.109 | 0.817 | 0.208 | 0.649 | 0.076 | 0.871 | 0.328 | 0.446 | 0.339 | 0.427 |
| | Event 3 | **0.370** | **0.638** | 0.706 | 0.309 | 1.131 | -0.107 | **0.533** | **0.478** | 0.553 | 0.458 | 0.872 | 0.146 | 0.752 | 0.264 | 1.192 | -0.167 | 1.323 | -0.295 |
| | **Test** | **0.005** | **0.937** | 0.008 | 0.904 | 0.015 | 0.811 | **0.006** | **0.927** | 0.006 | 0.925 | 0.011 | 0.855 | 0.008 | 0.898 | 0.015 | 0.811 | 0.016 | 0.795 |
| STOCKSFIELD | Event 1 | **0.019** | **0.747** | 0.047 | 0.389 | 0.879 | -0.130 | 0.279 | 0.642 | **0.250** | **0.678** | 0.568 | 0.270 | 0.443 | 0.430 | -1.01 | -0.299 | 1.097 | -0.410 |
| | Event 2 | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × |
| | Event 3 | 0.396 | 0.315 | **0.361** | **0.370** | 0.607 | -0.051 | **0.358** | **0.381** | 0.698 | -0.209 | 0.442 | 0.234 | 0.486 | 0.158 | 0.673 | -0.167 | 0.757 | -0.311 |
| | **Test** | **0.013** | **0.879** | 0.016 | 0.851 | 4.059 | -2.665 | **0.014** | **0.873** | 0.016 | 0.859 | 0.019 | 0.830 | 0.020 | 0.817 | 0.025 | 0.773 | 0.026 | 0.762 |
| NUNNYKIRK | Event 1 | **0.116** | **0.862** | 0.257 | 0.695 | 0.325 | 0.614 | 0.212 | 0.748 | **0.158** | **0.813** | 0.273 | 0.675 | 0.184 | 0.781 | 0.343 | 0.593 | 0.382 | 0.546 |
| | Event 2 | **0.043** | **0.926** | 0.056 | 0.902 | 0.249 | 0.566 | **0.054** | **0.907** | 0.282 | 0.509 | 0.133 | 0.768 | 0.093 | 0.839 | 0.180 | 0.686 | 0.216 | 0.624 |
| | Event 3 | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × |
| | **Test** | **0.003** | **0.972** | 0.004 | 0.958 | 0.009 | 0.925 | **0.004** | **0.962** | 0.005 | 0.951 | 0.006 | 0.944 | 0.005 | 0.954 | 0.009 | 0.923 | 0.009 | 0.922 |
| KNITSLEY | Event 1 | **0.008** | **0.960** | **0.017** | **0.910** | 0.160 | 0.164 | 0.029 | 0.845 | 0.037 | 0.808 | 0.122 | 0.362 | 0.027 | 0.856 | 0.148 | 0.224 | 0.143 | 0.251 |
| | Event 2 | **0.056** | **0.907** | 0.089 | 0.854 | 0.473 | 0.219 | **0.059** | **0.901** | 0.135 | 0.777 | 0.323 | 0.466 | 0.178 | 0.707 | 0.421 | 0.306 | 0.405 | 0.332 |
| | Event 3 | 0.028 | 0.738 | **0.017** | **0.839** | 0.091 | 0.168 | 0.021 | 0.803 | **0.012** | **0.890** | 0.072 | 0.299 | 0.033 | 0.697 | 0.092 | 0.152 | 0.093 | 0.147 |
| | **Test** | **0.004** | **0.939** | 0.004 | 0.928 | 0.012 | 0.810 | **0.003** | **0.942** | 0.004 | 0.930 | 0.008 | 0.862 | 0.005 | 0.911 | 0.011 | 0.821 | 0.011 | 0.824 |
| KIELDER | Event 1 | **0.008** | **0.957** | 0.015 | 0.920 | 0.140 | 0.269 | 0.016 | 0.918 | **0.013** | **0.933** | 0.091 | 0.527 | 0.031 | 0.837 | 0.137 | 0.286 | 0.123 | 0.361 |
| | Event 2 | **0.027** | **0.877** | 0.029 | 0.869 | 0.087 | 0.610 | 0.017 | 0.700 | **0.015** | **0.934** | 0.081 | 0.637 | 0.047 | 0.788 | 0.068 | 0.692 | 0.059 | 0.735 |
| | Event 3 | **0.013** | **0.691** | 0.015 | 0.634 | 0.040 | 0.280 | **0.019** | **0.668** | 0.021 | 0.629 | 0.021 | 0.621 | 0.023 | 0.618 | 0.040 | 0.284 | 0.042 | 0.260 |
| | **Test** | **0.003** | **0.962** | 0.004 | 0.942 | 0.014 | 0.826 | **0.004** | **0.951** | 0.004 | 0.946 | 0.009 | 0.894 | 0.005 | 0.940 | 0.013 | 0.844 | 0.013 | 0.845 |
| **Best** (↑) | Count | 16 | 16 | 0 | 0 | 0 | 0 | 4 | 4 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

### 4.6 MAIN RESULTS

Across six UK catchments and three events per site, APILANET achieves the strongest overall performance (Table 4). On the **Test** split it achieves the lowest MSE↓ and highest NSE↑ on *five out of six* catchments, with a very close second place on KNITSLEY (0.004/0.939 vs. 0.003/0.942 for TSMIXER). Aggregating over all event–level and test rows, APILANET secures **16** best scores, compared with **4** for TSMIXER and **2** for PATCHMIXER, while the remaining baselines never dominate. The largest gains are observed at ACOMB MFS, NUNNYKIRK and KIELDER, where APILANET consistently improves both error (MSE) and efficiency (NSE) over the strongest deep-learning baselines, indicating that the latent-physics prior is beneficial across a range of single-sensor catchment regimes.

## 5 CONCLUSION AND FUTURE WORK

We introduced APILANET, an **A**daptive **P**hysics-**I**nformed **La**tent **N**etwork for single-sensor forecasting that couples sequence learning with weak-form conservation. A dual-stream latent prior with input-driven gating, a monotone observation link, and a learned, normalized space–time measure deliver stable training and targeted physics enforcement. On five UK catchments, APILANET improves NSE and lowers MSE during extreme events over strong state-of-the-arts, suggesting a practical application for conservation-governed forecasting under sparse sensing.

We analyzed the limitations of our work and briefly discuss some directions for future research: (i) *Beyond 1-D.* Generalize the latent PDE from a reach-averaged 1-D mesh to multi-reach/graph geometries and lightweight momentum terms. (ii) *Safer observation mapping.* Add physics-aware shape priors and uncertainty quantification to the monotone link for robust extrapolation outside the observed latent range. (iii) *Richer general states and interpretability.* Learn time–space wetness/state variables (beyond a single decay $\kappa$) and integrate XAI diagnostics to attribute predictions to latent physics and drivers.

## ACKNOWLEDGMENTS

To preserve double-blind review, acknowledgments and funding details are intentionally omitted. They will be added in the camera-ready version upon acceptance.

## REFERENCES

Abdus Samad Azad, Nahina Islam, Md Nurun Nabi, Hifsa Khurshid, and Mohammad Ashraful Siddique. Developments and trends in water level forecasting using machine learning models—a review. *IEEE Access*, 13:63048–63065, 2025. doi: 10.1109/ACCESS.2025.3557910.

Cristian Challu, Kin G. Olivares, Boris N. Oreshkin, Federico Garza, Max Mergenthaler-Canseco, and Artur Dubrawski. N-hits: Neural hierarchical interpolation for time series forecasting, 2022. URL https://arxiv.org/abs/2201.12886.

Si-An Chen, Chun-Liang Li, Nate Yoder, Sercan O. Arik, and Tomas Pfister. Tsmixer: An all-mlp architecture for time series forecasting, 2023. URL https://arxiv.org/abs/2303.06053.

Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *International Conference on Machine Learning (ICML)*, 2024.

Zeying Gong, Yujin Tang, and Junwei Liang. Patchmixer: A patch-mixing architecture for long-term time series forecasting. 2023. URL https://api.semanticscholar.org/CorpusID:263334059.

E. Kharazmi, Z. Zhang, and G. E. Karniadakis. Variational physics-informed neural networks for solving partial differential equations, 2019. URL https://arxiv.org/abs/1912.00873.

Jungeun Kim, Kookjin Lee, Dongeun Lee, Sheo Yon Jhin, and Noseong Park. Dpm: A novel training method for physics-informed neural networks in extrapolation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(9):8146–8154, May 2021. doi: 10.1609/aaai.v35i9.16992. URL https://ojs.aaai.org/index.php/AAAI/article/view/16992.

Hang Jung Ling, Salomé Bru, Julia Puig, Florian Vixège, Simon Mendez, Franck Nicoud, Pierre-Yves Courand, Olivier Bernard, and Damien Garcia. Physics-guided neural networks for intraventricular vector flow mapping. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 71(11):1377–1388, 2024. doi: 10.1109/TUFFC.2024.3411718.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*, 2023.

Levi D. McClenny and Ulisses M. Braga-Neto. Self-adaptive physics-informed neural networks. *Journal of Computational Physics*, 474:111722, 2023. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2022.111722. URL https://www.sciencedirect.com/science/article/pii/S0021999122007859.

Juan Nathaniel, Yongquan Qu, Tung Nguyen, Sungduk Yu, Julius Busecke, Aditya Grover, and Pierre Gentine. Chaosbench: A multi-channel, physics-based benchmark for subseasonal-to-seasonal climate prediction. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 43715–43729. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/4d3684dd7926754b48bc6cd99a840232-Paper-Datasets_and_Benchmarks_Track.pdf.

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

Boris N. Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=r1ecqn4YwB.

M. Raissi, P. Perdikaris, and G.E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2018.10.045. URL https://www.sciencedirect.com/science/article/pii/S0021999118307125.

Franz M. Rohrhofer, Stefan Posch, Clemens Gößnitzer, and Bernhard C. Geiger. Data vs. physics: The apparent pareto front of physics-informed neural networks. *IEEE Access*, 11:86252–86261, 2023. doi: 10.1109/ACCESS.2023.3302892.

Tim De Ryck, Siddhartha Mishra, and Roberto Molinaro. wpinns: Weak physics informed neural networks for approximating entropy solutions of hyperbolic conservation laws, 2022. URL https://arxiv.org/abs/2207.08483.

Lingchen Wang, Tao Yang, and Bo Hu. A battery state-of-health estimation method for real-world electric vehicles based on physics-informed neural networks. *IEEE Sensors Journal*, 25 (9):15577–15587, 2025. doi: 10.1109/JSEN.2025.3549486.

Jeremy Yu, Lu Lu, Xuhui Meng, and George Em Karniadakis. Gradient-enhanced physics-informed neural networks for forward and inverse pde problems. *Computer Methods in Applied Mechanics and Engineering*, 393:114823, April 2022. ISSN 0045-7825. doi: 10.1016/j.cma.2022.114823. URL http://dx.doi.org/10.1016/j.cma.2022.114823.

Andrea Zanella, Sergio Zubelzu, and Mehdi Bennis. Sensor networks, data processing, and inference: The hydrology challenge. *IEEE Access*, 11:107823–107842, 2023. doi: 10.1109/ACCESS.2023.3318739.

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

# A Appendix A

**Ethics Statement** Kharazmi et al. (2019) Yu et al. (2022) Ryck et al. (2022) We used large language models (LLMs) solely to polish writing e.g., improving clarity, grammar, and flow. All ideas, methods, experiments, analyses, figures, and conclusions are the authors' own. No data, code, or results were generated by LLMs, and all citations and factual statements were verified by the authors.

**Reproducibility Statement** We provide the theoretical background throughout the paper and in the Technical Appendix, including assumptions, definitions, and proofs supporting our claims. Upon acceptance, we will release the full codebase, configuration files, and scripts to reproduce all experiments in a public GitHub repository; the URL will be announced to preserve double-blind review.

## A.1 Datasets

**Data source.** All datasets used in this study were extracted from the UK Environment Agency Hydrology service (https://environment.data.gov.uk/hydrology/explore). We used publicly available gauge series and constructed train/test splits per catchment as summarized in Table 5.

Table 5: Dataset overview by site (Train+Test merged). All series are 15 min cadence and include 10 features per site. Source: UK Environment Agency Hydrology.

| Site | Rows (total) | Features | Time range | Med. interval |
|---|---|---|---|---|
| Acomb GH | 320590 | 10 | 2016-01-01 — 2025-02-28 | 15 min |
| Acomb MSFD | 321260 | 10 | 2016-01-01 — 2025-02-28 | 15 min |
| Knitlsey | 315535 | 10 | 2016-01-01 — 2024-12-30 | 15 min |
| Kielder | 315525 | 10 | 2016-01-01 — 2024-12-30 | 15 min |
| Nunnykirk | 315505 | 10 | 2016-01-01 — 2024-12-30 | 15 min |
| Stocksfield | 110857 | 10 | 2022-01-01 — 2025-02-28 | 15 min |

**Preprocessing.** Timestamps were parsed and sorted; all series operate at a 15 min cadence. We retain provider units and engineer a 10D feature vector per timestamp. Here $\Delta h$ and $\Delta^2 h$ are first/second differences of level; `daily_min`/`daily_max` are previous-day extrema (computed per calendar day and shifted by 96 steps = 24 h to avoid leakage), then forward/backward filled; `future_rain` is a 32–step (8 h) lead of rain (placeholder when not observed); `AWI` is an exponentially weighted antecedent wetness index with 5-day decay; and `rain_3h`/`rain_24h` are rolling rainfall sums over 12 and 96 steps. After feature construction we drop any residual NaNs. Features are scaled with a Min–Max transform fitted on the training split and applied to validation/test. For sequence modeling we form input/output windows of 32/32 steps (8 h/8 h); training uses an 80/20 chronological split with shuffling only on the training loader (validation/test are not shuffled).

**Notation.** Let $\{t_\tau\}_{\tau=1}^{T}$ be the forecast timestamps (uniform step $\Delta t$), and let $y_\tau$ and $\hat{y}_\tau$ denote the observed and predicted water level at $t_\tau$.

**Mean Squared Error (MSE).**

$$\mathrm{MSE} \;=\; \frac{1}{T}\sum_{\tau=1}^{T}\big(\hat{y}_\tau - y_\tau\big)^2.$$

**Nash–Sutcliffe Efficiency (NSE).**

$$\mathrm{NSE} \;=\; 1 \;-\; \frac{\sum_{\tau=1}^{T}\big(\hat{y}_\tau - y_\tau\big)^2}{\sum_{\tau=1}^{T}\big(y_\tau - \bar{y}\big)^2}, \qquad \bar{y} \;=\; \frac{1}{T}\sum_{\tau=1}^{T} y_\tau.$$

**Peak timing error ($\Delta t_{\mathrm{peak}}$).** Let $\tau^\star_{\mathrm{obs}} \in \arg\max_\tau y_\tau$ and $\tau^\star_{\mathrm{pred}} \in \arg\max_\tau \hat{y}_\tau$. We report the (absolute) timing difference in hours:

$$\Delta t_{\mathrm{peak}} \;=\; \big|\, t_{\tau^\star_{\mathrm{pred}}} - t_{\tau^\star_{\mathrm{obs}}} \,\big| \;=\; \big|\, \tau^\star_{\mathrm{pred}} - \tau^\star_{\mathrm{obs}} \,\big|\, \Delta t.$$

(With 15 min cadence, $\Delta t = 0.25$ h.)

**Peak height error ($\Delta h_{\mathrm{peak}}$).** We compare the peak magnitudes over the forecast window:

$$\Delta h_{\mathrm{peak}} \;=\; \Big|\, \max_\tau \hat{y}_\tau \;-\; \max_\tau y_\tau \,\Big| \quad \text{(meters)}.$$

**Optimization & training.** All experiments are conducted on a single workstation with an NVIDIA RTX 4090 (24 GB), an Intel Core i9-14900KS, and 128 GB of RAM.[1] All models are trained in PyTorch with **Adam** (learning rate $1 \times 10^{-3}$), mini–batches of **64**, and shuffled training streams; validation/test loaders are not shuffled. We use a **deep ensemble** of $M{=}3$ independently trained instances for each seed we reinstantiate the data loaders with the same seed to obtain reproducible shuffles. At inference, we average ensemble outputs for the point forecast and report the ensemble standard deviation as an estimate of epistemic uncertainty. Unless otherwise stated, input and forecast horizons are 32 steps (15 min cadence $\Rightarrow$ 8 h lookback/8 h horizon), and the same preprocessing and scaling are applied across all runs.

[1] No multi-GPU or distributed training is used.

**Reproducibility.** We will release scripts that (i) download the raw CSVs from the Hydrology service, (ii) apply the exact parsing and split logic used in this paper, and (iii) regenerate all summary tables.

# B Appendix B : Panel A: Dual–Stream Discharge Prior with Input–Driven Gating

**Notation.** For a sequence $z \in \mathbb{R}^T$ define the forward difference $\Delta z(\tau) = z(\tau) - z(\tau-1)$ for $\tau \geq 2$. We write the Sobolev–seminorm $\|z\|_{\mathrm{H}^1}^2 = \sum_{\tau=2}^{T} (\Delta z(\tau))^2$ and the total variation $\|z\|_{\mathrm{TV}} = \sum_{\tau=2}^{T} |\Delta z(\tau)|$. A history window is $X_{1:L} \in \mathbb{R}^{L\times d}$; the most recent vector is $x_L \in \mathbb{R}^d$.

## B.1 Model and training objective

Two sequence encoders (e.g., LSTMs) produce nonnegative discharge sequences

$$Q_{\mathrm{b}}(X),\, Q_{\mathrm{q}}(X) \in \mathbb{R}^T_{\geq 0}, \qquad Q_{\mathrm{b}} = \phi_{\mathrm{b}}(X),\;\; Q_{\mathrm{q}} = \phi_{\mathrm{q}}(X),$$

and a scalar *gate* is computed from the history (in code: from $x_L$)

$$\alpha(X) = \sigma\big(g(X)\big) \in [0,1], \qquad \sigma(u) = \tfrac{1}{1+e^{-u}}.$$

The latent discharge propagated downstream is the convex mixture

$$Q_\theta(\tau; X) = \alpha(X)\, Q_{\mathrm{q}}(\tau; X) + \big(1-\alpha(X)\big)\, Q_{\mathrm{b}}(\tau; X), \qquad Q_\theta \in \mathbb{R}^T_{\geq 0}. \tag{12}$$

To bias the decomposition toward interpretable dynamics we add a paired prior

$$\mathcal{R}_{\mathrm{b}}(Q_{\mathrm{b}}) = \|Q_{\mathrm{b}}\|_{\mathrm{H}^1}^2, \qquad \mathcal{R}_{\mathrm{q}}(Q_{\mathrm{q}}) = \|Q_{\mathrm{q}}\|_{\mathrm{TV}}. \tag{13}$$

Let $\mathcal{L}_{\text{data}}$ denote the supervised loss (on the task outputs). The Panel-A contribution to the training objective is

$$\mathcal{L}_A(X;\theta) = \rho_{\mathrm{b}}\, \|Q_{\mathrm{b}}(X)\|_{\mathrm{H}^1}^2 + \rho_{\mathrm{q}}\, \|Q_{\mathrm{q}}(X)\|_{\mathrm{TV}}, \qquad \rho_{\mathrm{b}}, \rho_{\mathrm{q}} > 0, \tag{14}$$

and the full loss is $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{data}} + \mathcal{L}_A + \mathcal{L}_{\text{physics}}$.

**Remark (penalized joint learning).** Unlike a constrained "recover $(Q_{\mathrm{b}}, Q_{\mathrm{q}})$ given $Q_\theta$" solve, our implementation *jointly learns* $Q_{\mathrm{b}}, Q_{\mathrm{q}}$ with the encoders by penalizing equation 13 during training. This is exactly what the code does.

## B.2 Stability of the gated mixture

**Assumption B1 (encoder and gate regularity).** There exist Lipschitz constants $L_{\mathrm{b}}, L_{\mathrm{q}}, L_g \geq 0$ such that

$$\|Q_{\mathrm{b}}(X) - Q_{\mathrm{b}}(X')\|_\infty \leq L_{\mathrm{b}}\, \|X - X'\|, \quad \|Q_{\mathrm{q}}(X) - Q_{\mathrm{q}}(X')\|_\infty \leq L_{\mathrm{q}}\, \|X - X'\|,$$

and $|g(X) - g(X')| \leq L_g\, \|X - X'\|$, for a fixed norm $\|\cdot\|$ on $\mathbb{R}^{L\times d}$. We use the standard bound $|\sigma(u) - \sigma(v)| \leq \frac{1}{4}|u - v|$.

**Theorem 4** (Lipschitz dependence of $Q_\theta$ on the history)**.** *Under Assumption B1, for any windows $X, X'$,*

$$\big\|Q_\theta(\cdot; X) - Q_\theta(\cdot; X')\big\|_\infty \;\leq\; \Big(L_{\mathrm{q}} + L_{\mathrm{b}} + \tfrac{1}{4}\, L_g\, \Delta_Q(X')\Big)\, \|X - X'\|,$$

*where $\Delta_Q(X') = \sup_\tau \big|Q_{\mathrm{q}}(\tau; X') - Q_{\mathrm{b}}(\tau; X')\big|$. If a uniform bound $\Delta_Q(X') \leq \Delta_{\max}$ holds on the training domain, we may replace $\Delta_Q(X')$ by $\Delta_{\max}$.*

*Sketch.* Using equation 12,

$$\begin{aligned} Q_\theta(\cdot; X) - Q_\theta(\cdot; X') = {} & \alpha(X)\big(Q_{\mathrm{q}}(X) - Q_{\mathrm{q}}(X')\big) + \big(1-\alpha(X)\big)\big(Q_{\mathrm{b}}(X) - Q_{\mathrm{b}}(X')\big) \\ & + \big(\alpha(X) - \alpha(X')\big)\big(Q_{\mathrm{q}}(X') - Q_{\mathrm{b}}(X')\big). \end{aligned}$$

Take $\|\cdot\|_\infty$, apply the encoder Lipschitz bounds to the first two terms, and the sigmoid bound $|\alpha(X) - \alpha(X')| \leq \frac{1}{4}|g(X) - g(X')| \leq \frac{1}{4} L_g \|X - X'\|$ to the gate term; then collect constants.

**Interpretation.** Small perturbations of the input history yield bounded changes in $Q_\theta$. The bound decomposes additively into (i) variability of the fast stream, (ii) variability of the slow stream, and (iii) gate sensitivity scaled by the instantaneous separation $\Delta_Q$ between streams.

### B.3 Bias and identifiability of the penalized split

Define the per-batch objective

$$\mathcal{J}(X;\theta) = \mathcal{L}_{\text{data}}(X;\theta) + \rho_{\text{b}} \left\|Q_{\text{b}}(X)\right\|_{\text{H}^1}^2 + \rho_{\text{q}} \left\|Q_{\text{q}}(X)\right\|_{\text{TV}}.$$

At any stationary point of $\mathcal{J}$ (with respect to encoder parameters), the Euler–Lagrange/KKT conditions yield the following qualitative structure.

**Proposition 2** (Directional bias of the streams)**.** *Let $\theta^\star$ be a stationary point of $\mathcal{J}$. Then the slow stream $Q_{\text{b}}(X;\theta^\star)$ minimizes a data-augmented functional that contains $\|DQ\|_2^2$, while the fast stream $Q_{\text{q}}(X;\theta^\star)$ minimizes a data-augmented functional that contains $\|DQ\|_1$. Consequently, $Q_{\text{b}}$ concentrates low-frequency energy and $Q_{\text{q}}$ concentrates high-variation energy (sparse differences). The nonnegativity constraints preserve the physical sign.*

*Idea.* Differentiate $\mathcal{J}$ with respect to the encoder outputs. The gradient contributions of $\|Q_{\text{b}}\|_{\text{H}^1}^2$ and $\|Q_{\text{q}}\|_{\text{TV}}$ are, respectively, $D^\top(2\,DQ_{\text{b}})$ (a smoothing operator) and $D^\top(\text{sign}(DQ_{\text{q}}))$ (an edge-sparsifying operator). Balancing these with the data gradient yields the stated bias. Formal details follow by standard subdifferential calculus for TV.

**Identifiability discussion.** When $\alpha \in (0,1)$ and the two priors are active ($\rho_{\text{b}}, \rho_{\text{q}} > 0$), the optimization favors a unique *role allocation*—smooth content in $Q_{\text{b}}$, jump-sparse content in $Q_{\text{q}}$. If $\alpha$ saturates at $\{0,1\}$, the inactive stream is under-determined by the mixture; in practice we discourage saturation by ordinary early-training regularization on the gate (e.g., mild logit penalty) and by the data loss coupling both streams through $Q_\theta$.

## C Appendix C : Panel B: Properties of the Monotone Latent Mapping

Panel B maps the nonnegative driver $q(\tau) \in \mathbb{R}_{\geq 0}$ (output of Panel A) to the target $h(\tau)$ through a shallow MLP $f_\theta : \mathbb{R}_{\geq 0} \to \mathbb{R}$ applied elementwise in time: $h(\tau) = f_\theta\big(q(\tau)\big)$. We do *not* impose weight sign constraints; instead we add a lightweight *batchwise monotonicity surrogate* that encourages $f_\theta$ to be nondecreasing over the *observed* driver range.

Given a finite design set $\mathbf{q} = \{q_i\}_{i=1}^n$ sampled from the current batch (or a fixed grid) and sorted $q_{(1)} \leq \cdots \leq q_{(n)}$, define

$$\mathcal{L}_{\text{mono}}(\theta;\mathbf{q}) = \frac{1}{n-1} \sum_{i=1}^{n-1} \big[f_\theta(q_{(i+1)}) - f_\theta(q_{(i)})\big]_-, \qquad [x]_- = \max\{0,-x\}. \tag{15}$$

We add $\gamma_{\text{mono}}\mathcal{L}_{\text{mono}}$ to the training objective (with $\gamma_{\text{mono}}$=0.01 in our experiments).

**Proposition 3** (Immediate properties)**.** *If $f_\theta(q_{(i+1)}) \geq f_\theta(q_{(i)})$ for all $i$, then $\mathcal{L}_{\text{mono}}(\theta;\mathbf{q}) = 0$. Moreover,*

$$\max_{1\leq i\leq n-1} \left[f_\theta(q_{(i)}) - f_\theta(q_{(i+1)})\right]_+ \;\leq\; (n-1)\,\mathcal{L}_{\text{mono}}(\theta;\mathbf{q}),$$

*so the loss controls the largest adjacent monotonicity violation on the sampled range.*

Let design sets $\mathbf{q}^{(m)} \subset [0, Q_{\max}]$ densify (mesh size $\to 0$), and suppose $\sup_m \|f_{\theta_m}\|_\infty < \infty$ and a standard regularizer yields a uniform total-variation bound on $f_{\theta_m}$. If $\mathcal{L}_{\text{mono}}(\theta_m;\mathbf{q}^{(m)}) \to 0$, then a subsequence of $\{f_{\theta_m}\}$ converges pointwise a.e. on $[0, Q_{\max}]$ to a nondecreasing limit. *(Sketch:* Helly selection on uniformly BV functions + vanishing adjacent violations on a dense mesh implies monotonicity a.e. of the limit.*)*

**Practice.** (i) We form $\mathbf{q}$ by sorting the per-batch driver values and compute equation 15. (ii) The surrogate only constrains the map where data lie (observed driver range), which is sufficient to stabilize training and improve identifiability in practice. (iii) No architectural monotonicity constraints are required; the approach is optimizer- and MLP-agnostic.

# D APPENDIX D : PANEL C: WEAK-FORM PHYSICS ON A LATENT MESH

**Latent mesh and broadcasted residual.** Let the forecast steps be $\tau = 1{:}T$ and the latent spatial grid $\{x_j\}_{j=1}^{X} \subset [0,1]$. The model outputs two *time-indexed* proxies (constant in $x$ upon broadcast)

$$d_t h_\theta[\tau] \;\approx\; \partial_t h(\tau,\cdot), \qquad d_x Q_\theta[\tau] \;\approx\; \partial_x Q(\tau,\cdot),$$

and forms a latent forcing by projecting a single exogenous series via an exponential kernel

$$R_\kappa(x) \;=\; \bar{R}\, e^{-\kappa x}, \qquad \kappa > 0 \text{ learnable}, \;\; \bar{R} = \text{batch summary of rainfall}.$$

A nonnegative space–time weighting field $\lambda_\phi(\tau, x) \geq 0$ (produced by a small network on $(\tau, x)$) emphasizes informative regions. The broadcast weak residual is

$$r_\theta[\tau, j] \;=\; d_t h_\theta[\tau] \;+\; d_x Q_\theta[\tau] \;-\; R_\kappa(x_j),$$

and the weak-form physics loss used in training is the normalized weighted average

$$\mathcal{L}_{\text{pde}}(\theta,\phi) \;=\; \frac{1}{TX} \sum_{\tau=1}^{T} \sum_{j=1}^{X} \lambda_\phi(\tau, x_j)\, r_\theta[\tau, j]^2, \qquad \lambda_\phi(\tau, x) \geq 0. \tag{16}$$

(Implementation: $\lambda_\phi$ is Softplus-positive; optionally we renormalize it per batch so its average over $(\tau, j)$ is 1, but this is not required.)

## C.1 FROM CLASSICAL WEAK RESIDUALS TO THE BROADCAST LOSS

Consider the 1-D continuity law on a strip,

$$\partial_t h(\tau, x) + \partial_x Q(\tau, x) \;=\; R(x), \qquad (\tau, x) \in \{1{:}T\} \times [0,1].$$

Let $\mu_\phi$ be a learned *nonnegative* measure on $[0,1]$ with density $\lambda_\phi(\tau,\cdot)$ for each $\tau$ (no sign changes; boundedness holds in practice due to Softplus outputs).

**Theorem 5** (Broadcast loss is a weighted weak residual)**.** *Assume (i) $d_t h_\theta[\tau]$ and $d_x Q_\theta[\tau]$ are broadcast as piecewise-constant in $x$, (ii) $R_\kappa$ is continuous in $x$, and (iii) $\lambda_\phi(\tau,\cdot)$ is bounded and nonnegative. Then equation 16 is a Riemann (cell-wise) quadrature of the weighted weak residual with constant test functions on each cell:*

$$\mathcal{L}_{\text{pde}}(\theta,\phi) \;=\; \frac{1}{T} \sum_{\tau=1}^{T} \int_0^1 \big(\partial_t h_\theta(\tau, x) + \partial_x Q_\theta(\tau, x) - R_\kappa(x)\big)^2 \, \mathrm{d}\mu_\phi(\tau, x) \;+\; o(1),$$

*where $o(1) \to 0$ as $\max_j |x_{j+1} - x_j| \to 0$.* Sketch. *Broadcasting makes trial/test functions piecewise constant in $x$; the double sum is a normalized quadrature of the weighted $L^2$ residual over the latent cells.*

## C.2 CONSISTENCY UNDER REFINEMENT AND APPROXIMATION

We formalize when vanishing broadcast loss enforces the PDE almost everywhere.

**Assumption 5** (Approximation + bounded weights)**.** *There exist $h^\star, Q^\star, R^\star$ with $\partial_t h^\star + \partial_x Q^\star = R^\star$ a.e. such that: (i) $d_t h_\theta \to \partial_t h^\star$ and $d_x Q_\theta \to \partial_x Q^\star$ in $L^2([0,1])$ (over $\tau$); (ii) $R_\kappa \to R^\star$ in $L^2([0,1])$ as $\kappa \to \kappa^\star$; (iii) the latent grid fill distance $\to 0$; (iv) for each $\tau$, $\lambda_\phi(\tau,\cdot)$ is bounded on $[0,1]$ (and optionally renormalized to unit mean).*

**Theorem 6** (Consistency of latent weak enforcement)**.** *Under Assumption 5, if $\mathcal{L}_{\text{pde}}(\theta,\phi) \to 0$ then*

$$\partial_t h^\star(\tau, x) + \partial_x Q^\star(\tau, x) = R^\star(x) \qquad \textit{for a.e. } (\tau, x) \in \{1{:}T\} \times [0,1].$$

Sketch. *By Theorem 5 the discrete loss converges to a weighted $L^2$ residual; bounded $\lambda_\phi$ and the $L^2$ approximations imply the residual tends to 0 in $L^2(\mu_\phi)$, hence vanishes a.e.*

### C.3 Role of the learned weight field and exponential forcing

**Learned importance map.** The nonnegative field $\lambda_\phi(\tau, x)$ in equation 16 lets the model allocate *physics pressure* to informative regions (e.g., transients or specific latent cells). Gradients take the form

$$\frac{\partial \mathcal{L}_{\text{pde}}}{\partial d_t h_\theta[\tau]} = \frac{2}{TX}\sum_j \lambda_\phi(\tau, x_j)\, r_\theta[\tau, j], \quad \frac{\partial \mathcal{L}_{\text{pde}}}{\partial d_x Q_\theta[\tau]} = \frac{2}{TX}\sum_j \lambda_\phi(\tau, x_j)\, r_\theta[\tau, j],$$

$$\frac{\partial \mathcal{L}_{\text{pde}}}{\partial \phi} = \frac{1}{TX}\sum_{\tau,j} r_\theta[\tau, j]^2\, \partial_\phi \lambda_\phi(\tau, x_j). \quad (17)$$

so cells with large residuals attract more weight until balanced by normalization/other losses.

**Exponential projection.** With $R_\kappa(x) = \bar{R}e^{-\kappa x}$ and $\kappa > 0$ learned, single-point exogenous input induces a *spatial* latent loading that decays with $x$, enabling spatiotemporal structure from a single time series while keeping the projection differentiable and stable.

### C.4 Relation to classical PINNs and weak–form PINNs (mathematical)

**Classical (strong-form) PINNs.** For a PDE $\mathcal{N}[u] = f$ on $[1{:}T] \times \Omega$, strong PINNs penalize pointwise residuals at collocation points:

$$\mathcal{L}_{\text{strong}}(\theta) = \frac{1}{N}\sum_{i=1}^{N} \big|\mathcal{N}[u_\theta](\tau_i, x_i) - f(\tau_i, x_i)\big|^2 \; + \text{(data/bc/ic)}.$$

They require spatial collocation $(\tau_i, x_i)$ and (via $\mathcal{N}$) generally involve higher-order derivatives of $u_\theta$.

**Weak–form (Galerkin) PINNs.** Fix test functions $\{\varphi_k\}_{k=1}^K$; the weak residual is

$$\mathcal{R}_{\text{weak}}(\theta; \varphi_k) = \int_\Omega \big(\mathcal{N}[u_\theta] - f\big)\, \varphi_k \, \mathrm{d}x, \qquad \mathcal{L}_{\text{weak}}(\theta) = \frac{1}{K}\sum_{k=1}^{K} \big|\mathcal{R}_{\text{weak}}(\theta; \varphi_k)\big|^2 \; + \text{(data/bc/ic)}.$$

With cellwise-constant $\varphi_k = \mathbb{1}_{\Omega_k}$ this becomes a per-cell *averaged* $L^2$ residual, trading pointwise sensitivity for integral robustness.

**APILaNet's broadcast weak form (Panel C).** On a *latent* 1-D grid $\{x_j\}_{j=1}^X$, we broadcast time-only proxies $d_t h_\theta[\tau]$ and $d_x Q_\theta[\tau]$ and use an exponentially projected forcing $R_\kappa(x) = \bar{R}e^{-\kappa x}$:

$$r_\theta[\tau, j] = d_t h_\theta[\tau] + d_x Q_\theta[\tau] - R_\kappa(x_j), \qquad \mathcal{L}_{\text{pde}}(\theta, \phi) = \frac{1}{TX}\sum_{\tau=1}^{T}\sum_{j=1}^{X} \lambda_\phi(\tau, x_j)\, r_\theta[\tau, j]^2,$$

with a learned nonnegative measure $\lambda_\phi(\tau, \cdot)$ (Sec. **??**). By Thm. 5, $\mathcal{L}_{\text{pde}}$ is a *Riemann quadrature* of a weighted weak $L^2$ residual with constant test functions.

## E Appendix E : Panel D: Properties and Pseudo-code

**Recall (from Method, Eqns. equation 9–equation 11).** The effective PDE weight factorizes as

$$\Lambda_{\text{pde}}(t, x) = \lambda_{\text{pde}}\, \lambda_{\text{loc}}(t, x), \quad \lambda_{\text{loc}}(t, x) \geq 0, \quad \frac{1}{TX}\sum_{\tau=1}^{T}\sum_{j=1}^{X} \lambda_{\text{loc}}(\tau, x_j) = 1,$$

and the PDE contribution to the loss is

$$\mathcal{L}_{\text{pde}}^{\text{eff}} = \lambda_{\text{pde}}\, \frac{1}{TX}\sum_{\tau=1}^{T}\sum_{j=1}^{X} \lambda_{\text{loc}}(\tau, x_j)\, r_\theta[\tau, j]^2, \quad r_\theta[\tau, j] = \partial_t h_\theta[\tau] + \partial_x Q_\theta[\tau] - R_\theta(x_j).$$

Global weights are scheduled per mini-batch $i \in \{\text{pde}, \text{cons}\}$ by

$$\lambda_i = \text{clip}\Big(\lambda_i^0\big(1 + E + \boldsymbol{\alpha}_i^\top \mathbf{s} + \alpha_{i,\Pi}\,\Pi\big), \lambda_i^{\min}, \lambda_i^{\max}\Big),$$

with base $\lambda_i^0 > 0$, nonnegative sensitivities $(\boldsymbol{\alpha}_i, \alpha_{i,\Pi})$, and clipping bounds.

### D.1 Assumptions and immediate consequences

**Assumption 6** (Bounded signals & normalized local field)**.** *During training the batch prediction loss $E \geq 0$, each component of the regime vector $\mathbf{s} \geq 0$, and the activity score $\Pi \in [0,1]$ are bounded. The local field obeys $\lambda_{loc}(\tau, x) \geq 0$ and $\frac{1}{TX}\sum_{\tau,j} \lambda_{loc}(\tau, x_j) = 1$. The clip enforces $\lambda_i \in [\lambda_i^{\min}, \lambda_i^{\max}]$.*

**Theorem 7** (Monotone responsiveness with bounded pressure)**.** *Under Assumption 6, each $\lambda_i$ is (piecewise) nondecreasing in $E$, in every component of $\mathbf{s}$, and in $\Pi$ (whenever unclipped), and always satisfies $\lambda_i^{\min} \leq \lambda_i \leq \lambda_i^{\max}$. Moreover, when unclipped,*

$$\frac{\partial \lambda_i}{\partial E} = \lambda_i^0, \qquad \frac{\partial \lambda_i}{\partial s_k} = \alpha_{ik}\lambda_i^0, \qquad \frac{\partial \lambda_i}{\partial \Pi} = \alpha_{i,\Pi}\lambda_i^0.$$

**Proposition 4** (Lipschitz variation across batches)**.** *For consecutive batches $k, k{+}1$, when unclipped*

$$\left|\lambda_i^{(k+1)} - \lambda_i^{(k)}\right| \leq \lambda_i^0 \left( |E_{k+1} - E_k| + \sum_m \alpha_{im} |s_{m,k+1} - s_{m,k}| + \alpha_{i,\Pi} |\Pi_{k+1} - \Pi_k| \right),$$

*and with clipping, the same bound holds after projection to $[\lambda_i^{\min}, \lambda_i^{\max}]$. Thus the scheduler is Lipschitz in signal deltas and has no EMA-type lag.*

**Lemma 1** (Scale invariance under local normalization)**.** *With $\frac{1}{TX}\sum_{\tau,j} \lambda_{loc}(\tau, x_j) = 1$,*

$$\mathcal{L}_{\text{pde}}^{eff} = \lambda_{\text{pde}} \cdot \overline{r^2}, \quad \overline{r^2} := \frac{1}{TX} \sum_{\tau,j} \lambda_{loc}(\tau, x_j)\, r_{\tau j}^2.$$

*Hence the rescaling $\lambda_{loc} \mapsto c\,\lambda_{loc}$, $\lambda_{\text{pde}} \mapsto \lambda_{\text{pde}}/c$ leaves $\mathcal{L}_{\text{pde}}^{eff}$ unchanged; normalization removes this ambiguity and improves identifiability.*

### D.2 Gradients and intuition

Using $r_{\tau j} = d_t h_\theta[\tau] + d_x Q_\theta[\tau] - R_\theta(x_j)$, the partials of $\mathcal{L}_{\text{pde}}^{\text{eff}}$ are

$$\begin{aligned}
\frac{\partial \mathcal{L}_{\text{pde}}^{\text{eff}}}{\partial d_t h_\theta[\tau]} &= \frac{2\lambda_{\text{pde}}}{TX} \sum_j \lambda_{\text{loc}}(\tau, x_j)\, r_{\tau j}, \\
\frac{\partial \mathcal{L}_{\text{pde}}^{\text{eff}}}{\partial d_x Q_\theta[\tau]} &= \frac{2\lambda_{\text{pde}}}{TX} \sum_j \lambda_{\text{loc}}(\tau, x_j)\, r_{\tau j}, \\
\frac{\partial \mathcal{L}_{\text{pde}}^{\text{eff}}}{\partial \lambda_{\text{loc}}(\tau, x_j)} &= \frac{\lambda_{\text{pde}}}{TX}\, r_{\tau j}^2 \quad \text{(before renormalization)}.
\end{aligned} \tag{18}$$

Thus the learned field $\lambda_{\text{loc}}$ (Softplus-positive) allocates more weight to large residuals until balanced by normalization and other losses; $\lambda_{\text{pde}}$ scales the overall physics pressure per batch.

### D.3 Pseudo-code (domain-agnostic)

We use the factorized schedule in Algorithm 2. It matches the Method section but is formatted for one column.

### D.4 Practical knobs

**Clips.** Choose $[\lambda_i^{\min}, \lambda_i^{\max}]$ so physics never dominates early but can rise during events. **Sensitivities.** Start with small $\alpha$s (e.g., $10^{-1}$–$10^{0}$), increase if residuals persist. **Spread regularizers (optional).** Entropy or $\ell_2$ penalties on $\lambda_{\text{loc}}$ discourage collapse:

$$\mathcal{R}_{\text{entropy}} = \beta \sum_{\tau,j} \lambda_{\text{loc}}(\tau, x_j) \log \lambda_{\text{loc}}(\tau, x_j), \quad \mathcal{R}_{\ell_2} = \beta \sum_{\tau,j} \left(\lambda_{\text{loc}}(\tau, x_j) - \tfrac{1}{X}\right)^2.$$

**Algorithm 2:** Adaptive Multi-Loss Scheduling with Factorized Local Weights

**Inputs:** mini-batch $\mathcal{D}$, model $\mathcal{F}_\theta$, optimizer; bases $\{\lambda_i^0\}$; sensitivities $\{\alpha_{ik}\}$; clips $[\lambda_i^{\min}, \lambda_i^{\max}]$
**Outputs:** updated parameters $\theta$
**for** *epoch* $e = 1$ **to** $N_{epoch}$ **do**
  **foreach** *mini-batch* $\mathcal{D}$ **do**
    compute per-losses $\{\mathcal{L}_i(\theta, \mathcal{D})\}_{i=1}^m$; optional local map $W_{\text{loc}} \geq 0$
    compute batch signals $\{s_k(\mathcal{D})\}_{k=1}^K$ and activity $\Pi$
    **for** $i = 1$ **to** $m$ **do**
      $\lambda_i \leftarrow \text{clip}\Big(\lambda_i^0\big(1 + \sum_{k=1}^{K} \alpha_{ik}\, s_k + \alpha_{i,\Pi}\Pi\big), \lambda_i^{\min}, \lambda_i^{\max}\Big)$
    **if** $W_{loc}$ *used* **then**
      $Z \leftarrow \frac{1}{|\Omega|} \sum_{(t,x)\in\Omega} W_{\text{loc}}(t,x)$;
      $W_{\text{loc}} \leftarrow W_{\text{loc}}/Z$
    $\mathcal{L}_{\text{tot}} \leftarrow \sum_{i=1}^{m} \lambda_i\, \mathcal{L}_i(\theta, \mathcal{D}; W_{\text{loc}})$
    optimizer.zero_grad();
    backprop($\mathcal{L}_{\text{tot}}$);
    optimizer.step()

# F Appendix F : Additional Experiments

Table 6: Sensitivity of APILaNet to the number of latent cells $X$ and the learned measure $\lambda_\phi(t, x)$ on the synthetic benchmark.

| $X$ | Measure | Test MSE | Test NSE | ΔMSE vs. uniform | ΔNSE vs. uniform |
|---|---|---|---|---|---|
| – | Uniform (all $X$) | $8.55 \times 10^{-4}$ | 0.9038 | – | – |
| 8 | Learned $\lambda_\phi$ | $8.49 \times 10^{-4}$ | 0.9044 | $\approx -0.7\%$ | $\approx +0.0006$ |
| 16 | Learned $\lambda_\phi$ | $7.01 \times 10^{-4}$ | 0.9210 | $\approx -18.0\%$ | $\approx +0.0172$ |
| 32 | Learned $\lambda_\phi$ | $7.26 \times 10^{-4}$ | 0.9183 | $\approx -15.1\%$ | $\approx +0.0145$ |
| 64 | Learned $\lambda_\phi$ | $8.30 \times 10^{-4}$ | 0.9066 | $\approx -2.9\%$ | $\approx +0.0028$ |

Table 6 summarizes the sensitivity of APILaNet to the number of latent cells $X$ and the learned weighting measure $\lambda_\phi(t, x)$. The uniform baseline aggregates performance across all $X$ with a fixed, non–adaptive measure, while the learned $\lambda_\phi$ is trained separately for each resolution $X \in \{8, 16, 32, 64\}$. Across all tested resolutions, the learned measure *never underperforms* the uniform baseline: the largest gains occur at moderate resolutions ($X = 16, 32$), with test MSE reduced by roughly 15–18% and NSE improved by about 0.015–0.017. For coarser or finer grids ($X = 8$ or 64), the gains are smaller but remain non–negative. This pattern indicates that APILaNet is not brittle with respect to the choice of latent discretization: performance varies smoothly around a favorable range of $X$, rather than collapsing for suboptimal resolutions.

We report additional benchmarks that stress early–warning skill at four lead times before the observed peak: **8 h**, **6 h**, **4 h**, and **2 h**. At each lead time we (i) re-slice the dataset around the peak time; (ii) run every model with the same hyperparameters as Section 4; and (iii) report the mean across three seeds. Primary metrics are MSE (↓) and NSE (↑); we additionally report peak timing error $\Delta t_{\text{peak}}$ (↓) and peak magnitude error $\Delta h_{\text{peak}}$ (↓). Across all sites, accuracy improves monotonically as lead time shortens (8 h→2 h). APILaNet retains the best or second-best MSE/NSE at every lead time and consistently reduces $\Delta t_{\text{peak}}$ and $\Delta h_{\text{peak}}$ relative to strong sequence baselines.

## F.1 Adaptive Physics Scheduler: Implementation and Sensitivity

For completeness, we restate the adaptive scheduler used in Panel D. The total loss is

$$\mathcal{L}_{\text{tot}} = \mathcal{L}_{\text{data}} + \lambda_{\text{pde}}\, \mathcal{L}_{\text{pde}} + \lambda_{\text{cons}}\, \mathcal{L}_{\text{cons}} + \lambda_{\text{mono}}\, \mathcal{L}_{\text{mono}}, \tag{19}$$

where $\lambda_{\text{pde}}$ and $\lambda_{\text{cons}}$ are adaptive global weights and $\lambda_{\text{mono}}$ is a small fixed coefficient.

Table 7: Sensitivity of the adaptive scheduler to the global physics scale $\lambda_{\text{scale}}$, the peak–sensitivity coefficient $\alpha_{\Pi}$, and the use of adaptive vs. static weights on the synthetic benchmark. Metrics are reported on the held–out test set.

| Experiment | $\lambda_{\text{scale}}$ | $\alpha_{\Pi}$ | Adaptive? | Test MSE ↓ | Test NSE ↑ |
|---|---|---|---|---|---|
| `lambda_scale_0.5` | 0.5 | 0.30 | Yes | 0.025706 | −0.271654 |
| `lambda_scale_1.0` | 1.0 | 0.30 | Yes | 0.013461 | 0.334087 |
| `lambda_scale_2.0` | 2.0 | 0.30 | Yes | 0.008709 | 0.569180 |
| `peak_coeff_0.00` | 1.0 | 0.00 | Yes | 0.015020 | 0.256971 |
| `peak_coeff_0.30` | 1.0 | 0.30 | Yes | 0.014446 | 0.285373 |
| `peak_coeff_0.60` | 1.0 | 0.60 | Yes | 0.013416 | 0.336320 |
| `no_adapt_static_lambda` | 1.0 | 0.30 | No | 0.012356 | 0.388777 |

Given the batch prediction loss $E \geq 0$, a vector of non–negative auxiliary signals $\mathbf{s} \in \mathbb{R}^K_{\geq 0}$, and an activity score $\Pi \in [0, 1]$, the global weights for $i \in \{\text{pde}, \text{cons}\}$ are updated instantaneously per mini–batch as

$$\lambda_i = \text{clip}\Big(\lambda_i^0\big(1 + E + \boldsymbol{\alpha}_i^\top \mathbf{s} + \alpha_{i,\Pi}\,\Pi\big),\ \lambda_i^{\min},\ \lambda_i^{\max}\Big), \tag{20}$$

where $\lambda_i^0 > 0$ is a base level, $(\boldsymbol{\alpha}_i, \alpha_{i,\Pi}) \geq 0$ are sensitivities, and clip enforces user–specified bounds $[\lambda_i^{\min}, \lambda_i^{\max}]$. The local field $\lambda_{\text{loc}}(t, x)$ is produced by a small network $A_\psi$ on normalized coordinates $(\tilde{t}, \tilde{x}) \in [0, 1]^2$,

$$\lambda_{\text{loc}}(\tau, x_j) = \frac{A_\psi(\tilde{t}_\tau, \tilde{x}_j)}{\frac{1}{TX}\sum_{\tau', j'} A_\psi(\tilde{t}_{\tau'}, \tilde{x}_{j'})}, \tag{21}$$

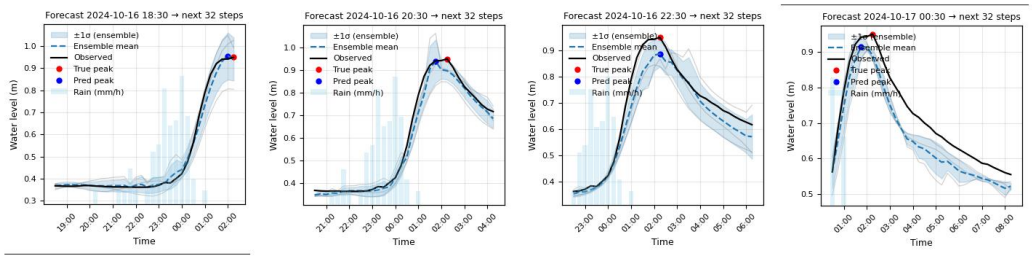
which guarantees the normalization property in equation **??**.

In all experiments we specify, for each $i \in \{\text{pde}, \text{cons}\}$, a base level $\lambda_i^0$, clipping bounds $(\lambda_i^{\min}, \lambda_i^{\max})$, and non–negative sensitivities $(\boldsymbol{\alpha}_i, \alpha_{i,\Pi})$. The only scalars selected by validation are a global physics scale $\lambda_{\text{scale}}$ (multiplying $(\lambda_{\text{pde}}^0, \lambda_{\text{cons}}^0)$) and an activity sensitivity $\alpha_\Pi$ applied to $\Pi$; we choose $(\lambda_{\text{scale}}, \alpha_\Pi)$ once by a small grid search on the validation NSE and reuse the same pair for all datasets within each benchmark.

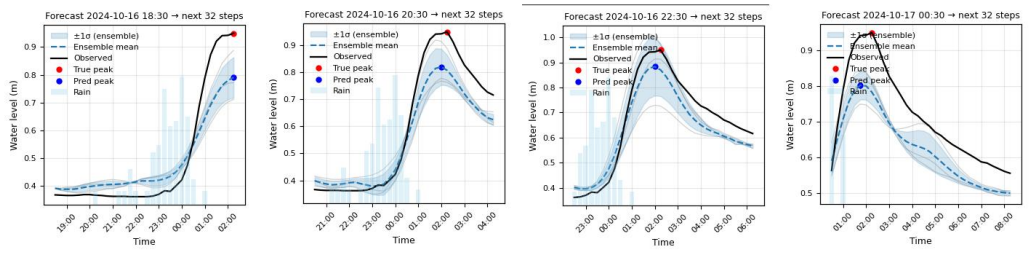
### F.2 Scheduler sensitivity study

To quantify robustness and provide the requested sensitivity analysis, we run a scheduler ablation on a synthetic single–sensor benchmark. We vary the global physics scale $\lambda_{\text{scale}} \in \{0.5, 1.0, 2.0\}$ and the peak–sensitivity coefficient $\alpha_\Pi \in \{0, 0.3, 0.6\}$, and compare adaptive ($\boldsymbol{\alpha}_i > 0$) versus static ($\boldsymbol{\alpha}_i = 0$) global weights. Test MSE and NSE on the held–out test set are reported in Table 7.

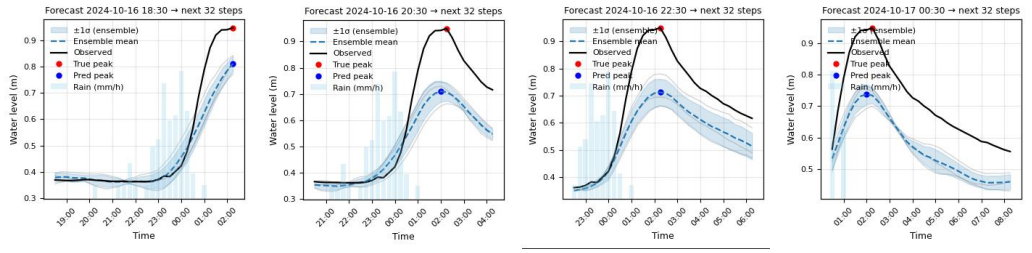
Across this grid, the scheduler behaves in a stable and smooth regime. Increasing $\lambda_{\text{scale}}$ from $0.5$ to $2.0$ strengthens the relative emphasis on physics and monotonically improves NSE (from $-0.27$ to $0.57$) without any training instabilities. Varying $\alpha_\Pi$ from $0$ to $0.6$ at fixed $\lambda_{\text{scale}} = 1.0$ yields only modest, smooth changes in the test performance, indicating that the scheduler does not rely on finely tuned coefficients. Finally, adaptive and static global weights achieve comparable overall NSE (roughly $0.33$ vs. $0.39$); the role of the adaptive scheduler is primarily to redistribute physics pressure towards difficult regimes (sharp transients and peaks), rather than to maximise aggregate error metrics.




(a) APILANET




(b) CROSSFORMER




(c) TSMIXER

Figure 4: Model forecasts at four start times: (a) APILANET, (b) CROSSFORMER, (c) TSMIXER.

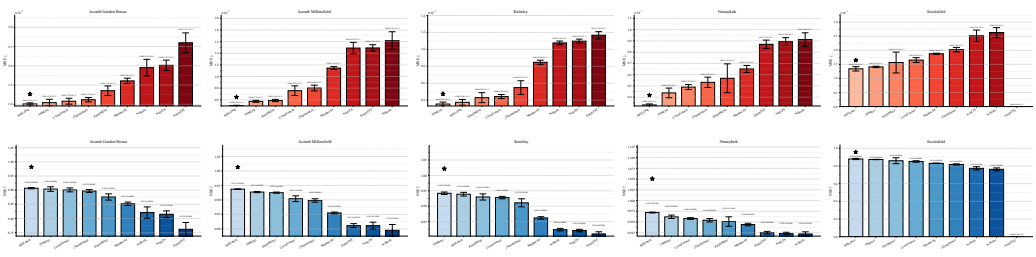


Figure 5: **Test performance across five UK catchments.** Bars show NSE (↑) and MSE (↓; $\times 10^{-3}$ axis units) for APILANET and baselines; error bars denote mean±SD over 3 seeds.

Table 8: Catchment-level forecasting 8 hours before peak. Metrics are mean±SD across seeds. Errors: peak timing $\Delta t_{\text{peak}}$ (h)↓, peak height $\Delta h_{\text{peak}}$ (m)↓, MSE↓, NSE↑.

| Data | Split | APILANET | | | | CROSSFORMER | | | | TSMIXER | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $\Delta t_{\text{peak}}$↓ | $\Delta h_{\text{peak}}$↓ | MSE↓ | NSE↑ | $\Delta t_{\text{peak}}$↓ | $\Delta h_{\text{peak}}$↓ | MSE↓ | NSE↑ | $\Delta t_{\text{peak}}$↓ | $\Delta h_{\text{peak}}$↓ | MSE↓ | NSE↑ |
| ACOMB GRN | Event 1 | 0.420 ± 0.380 | 0.299 ± 0.031 | 0.133 ± 0.096 | 0.623 ± 0.271 | 0.000 ± 0.000 | 0.377 ± 0.148 | 0.242 ± 0.09 | 0.314 ± 0.255 | 2.580 ± 4.470 | 0.552 ± 0.024 | 0.369 ± 0.032 | -0.044 ± 0.090 |
| | Event 2 | 0.170 ± 0.290 | 0.314 ± 0.055 | 0.198 ± 0.072 | 0.766 ± 0.085 | 0.250 ± 0.250 | 0.527 ± 0.007 | 0.479 ± 0.034 | 0.434 ± 0.041 | 0.500 ± 0.000 | 0.411 ± 0.043 | 0.354 ± 0.051 | 0.583 ± 0.060 |
| | Event 3 | 0.170 ± 0.290 | 1.339 ± 0.088 | 1.205 ± 0.185 | 0.132 ± 0.133 | 0.000 ± 0.000 | 1.348 ± 0.050 | 1.111 ± 0.837 | 0.200 ± 0.060 | 0.000 ± 0.000 | 1.297 ± 0.051 | 1.012 ± 0.089 | 0.271 ± 0.064 |
| | **Average** | 0.253 ± 0.144 | 0.651 ± 0.596 | 0.512 ± 0.601 | 0.507 ± 0.333 | 0.083 ± 0.144 | 0.751 ± 0.523 | 0.611 ± 0.449 | 0.316 ± 0.117 | 1.027 ± 1.368 | 0.753 ± 0.476 | 0.578 ± 0.376 | 0.270 ± 0.314 |
| ACOMB MFS | Event 1 | 0.000 ± 0.000 | 0.122 ± 0.065 | 0.064 ± 0.023 | 0.877 ± 0.044 | 0.000 ± 0.000 | 0.334 ± 0.098 | 0.201 ± 0.136 | 0.612 ± 0.262 | 0.000 ± 0.000 | 0.237 ± 0.030 | 0.112 ± 0.040 | 0.783 ± 0.078 |
| | Event 2 | 0.000 ± 0.000 | 0.107 ± 0.075 | 0.033 ± 0.010 | 0.877 ± 0.040 | 0.000 ± 0.000 | 0.192 ± 0.054 | 0.109 ± 0.054 | 0.586 ± 0.206 | 0.000 ± 0.000 | 0.101 ± 0.018 | 0.034 ± 0.015 | 0.870 ± 0.058 |
| | Event 3 | 0.000 ± 0.000 | 0.827 ± 0.045 | 0.665 ± 0.046 | 0.572 ± 0.029 | 0.000 ± 0.000 | 1.166 ± 0.062 | 1.267 ± 0.129 | 0.184 ± 0.083 | 0.000 ± 0.000 | 0.929 ± 0.079 | 0.805 ± 0.148 | 0.481 ± 0.096 |
| | **Average** | 0.000 ± 0.000 | 0.352 ± 0.411 | 0.254 ± 0.356 | 0.775 ± 0.176 | 0.000 ± 0.000 | 0.564 ± 0.526 | 0.526 ± 0.644 | 0.461 ± 0.240 | 0.000 ± 0.000 | 0.422 ± 0.444 | 0.317 ± 0.424 | 0.711 ± 0.204 |
| STOCKSFIELD | Event 1 | 0.000 ± 0.000 | 0.463 ± 0.192 | 0.452 ± 0.135 | 0.506 ± 0.147 | 2.080 ± 3.610 | 1.022 ± 0.052 | 1.072 ± 0.167 | -0.172 ± 0.182 | 0.080 ± 0.140 | 0.850 ± 0.065 | 0.689 ± 0.109 | 0.246 ± 119 |
| | Event 2 | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × |
| | Event 3 | 0.000 ± 0.000 | 0.949 ± 0.033 | 0.900 ± 0.068 | -0.077 ± 0.082 | 0.000 ± 0.000 | 0.995 ± 0.014 | 1.006 ± 0.039 | -0.203 ± 0.047 | 0.000 ± 0.000 | 0.971 ± 0.017 | 0.947 ± 0.036 | -0.133 ± 0.043 |
| | **Average** | 0.000 ± 0.000 | 0.471 ± 0.475 | 0.451 ± 0.450 | 0.143 ± 0.317 | 0.693 ± 1.201 | 0.672 ± 0.582 | 0.693 ± 0.601 | -0.125 ± 0.109 | 0.027 ± 0.046 | 0.607 ± 0.529 | 0.545 ± 0.490 | 0.038 ± 0.192 |
| NUNNYKIRK | Event 1 | 4.750 ± 4.160 | 0.241 ± 0.050 | 0.171 ± 0.089 | -0.762 ± 0.922 | 6.830 ± 0.880 | 0.189 ± 0.081 | 0.111 ± 0.019 | -0.145 ± 0.204 | 5.170 ± 4.470 | 0.246 ± 0.046 | 0.266 ± 0.157 | -1.744 ± 1.623 |
| | Event 2 | 0.000 ± 0.000 | 0.266 ± 0.059 | 0.295 ± 0.133 | 0.326 ± 0.305 | 0.000 ± 0.000 | 0.330 ± 0.088 | 0.278 ± 0.158 | 0.364 ± 0.361 | 0.000 ± 0.000 | 0.312 ± 0.090 | 0.302 ± 0.119 | 0.309 ± 0.274 |
| | Event 3 | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × |
| | **Average** | 1.583 ± 2.741 | 0.169 ± 0.147 | 0.155 ± 0.148 | -0.145 ± 0.558 | 2.277 ± 3.946 | 0.173 ± 0.166 | 0.130 ± 0.140 | 0.073 ± 0.262 | 1.723 ± 2.986 | 0.186 ± 0.164 | 0.189 ± 0.165 | -0.478 ± 1.107 |
| KNITSLEY | Event 1 | 0.170 ± 0.140 | 0.106 ± 0.033 | 0.028 ± 0.023 | 0.935 ± 0.053 | 0.000 ± 0.000 | 0.159 ± 0.090 | 0.079 ± 0.048 | 0.821 ± 0.109 | 0.000 ± 0.000 | 0.137 ± 0.036 | 0.073 ± 0.028 | 0.834 ± 0.064 |
| | Event 2 | 0.080 ± 0.140 | 0.155 ± 0.153 | 0.064 ± 0.024 | 0.916 ± 0.032 | 0.420 ± 0.720 | 0.317 ± 0.034 | 0.441 ± 0.168 | 0.429 ± 0.218 | 0.000 ± 0.000 | 0.287 ± 0.207 | 0.195 ± 0.172 | 0.748 ± 0.223 |
| | Event 3 | 0.000 ± 0.000 | 0.170 ± 0.008 | 0.124 ± 0.047 | 0.271 ± 0.274 | 0.000 ± 0.000 | 0.084 ± 0.013 | 0.047 ± 0.008 | 0.725 ± 0.050 | 0.000 ± 0.000 | 0.197 ± 0.019 | 0.099 ± 0.015 | 0.414 ± 0.090 |
| | **Average** | 0.083 ± 0.085 | 0.144 ± 0.033 | 0.072 ± 0.048 | 0.707 ± 0.378 | 0.140 ± 0.242 | 0.187 ± 0.119 | 0.189 ± 0.219 | 0.658 ± 0.204 | 0.000 ± 0.000 | 0.207 ± 0.075 | 0.122 ± 0.064 | 0.665 ± 0.222 |
| KIELDER | Event 1 | 0.000 ± 0.000 | 0.054 ± 0.041 | 0.011 ± 0.014 | 0.764 ± 0.292 | 0.000 ± 0.000 | 0.050 ± 0.027 | 0.027 ± 0.028 | -0.902 ± 1.990 | 0.000 ± 0.000 | 0.056 ± 0.004 | 0.016 ± 0.007 | 0.676 ± 0.163 |
| | Event 2 | 0.000 ± 0.000 | 0.081 ± 0.069 | 0.052 ± 0.063 | 0.071 ± 1.141 | 0.000 ± 0.000 | 0.082 ± 0.050 | 0.159 ± 0.109 | -3.057 ± 2.798 | 0.000 ± 0.000 | 0.101 ± 0.018 | 0.034 ± 0.015 | 0.870 ± 0.058 |
| | Event 3 | 1.420 ± 0.520 | 0.042 ± 0.048 | 0.016 ± 0.017 | 0.645 ± 0.386 | 1.820 ± 0.320 | 0.045 ± 0.050 | 0.018 ± 0.023 | 0.565 ± 0.055 | 1.420 ± 0.800 | 0.054 ± 0.017 | 0.046 ± 0.041 | -0.040 ± 0.922 |
| | **Average** | 0.473 ± 0.173 | 0.060 ± 0.053 | 0.026 ± 0.031 | 0.493 ± 0.606 | 0.606 ± 0.106 | 0.059 ± 0.042 | 0.068 ± 0.053 | -1.131 ± 1.614 | 0.473 ± 0.267 | 0.070 ± 0.013 | 0.032 ± 0.021 | 0.502 ± 0.381 |

Table 9: Catchment-level forecasting 6 hours before peak. Metrics are mean±SD across seeds. Errors: peak timing $\Delta t_{\text{peak}}$ (h)↓, peak height $\Delta h_{\text{peak}}$ (m)↓, MSE↓, NSE↑.

| Data | Split | APILANET | | | | CROSSFORMER | | | | TSMIXER | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $\Delta t_{\text{peak}}$↓ | $\Delta h_{\text{peak}}$↓ | MSE↓ | NSE↑ | $\Delta t_{\text{peak}}$↓ | $\Delta h_{\text{peak}}$↓ | MSE↓ | NSE↑ | $\Delta t_{\text{peak}}$↓ | $\Delta h_{\text{peak}}$↓ | MSE↓ | NSE↑ |
| ACOMB GRN | Event 1 | 0.750 ± 0.250 | 0.395 ± 0.073 | 0.351 ± 0.165 | 0.553 ± 0.210 | 0.250 ± 0.000 | 0.484 ± 0.151 | 0.447 ± 0.359 | 0.430 ± 0.459 | 0.830 ± 0.520 | 0.581 ± 0.088 | 0.665 ± 0.239 | 0.152 ± 0.305 |
| | Event 2 | 0.750 ± 0.500 | 0.351 ± 0.037 | 0.318 ± 0.096 | 0.564 ± 0.131 | 0.500 ± 0.430 | 0.462 ± 0.032 | 0.478 ± 0.075 | 0.345 ± 0.102 | 1.000 ± 0.430 | 0.344 ± 0.010 | 0.268 ± 0.069 | 0.632 ± 0.095 |
| | Event 3 | 1.580 ± 0.290 | 1.233 ± 0.122 | 4.814 ± 0.362 | 0.082 ± 0.069 | 1.580 ± 0.140 | 1.339 ± 0.089 | 4.562 ± 0.497 | 0.130 ± 0.095 | 1.250 ± 0.500 | 1.320 ± 0.074 | 4.171 ± 0.413 | 0.205 ± 0.079 |
| | **Average** | 1.027 ± 0.479 | 0.660 ± 0.497 | 1.828 ± 2.586 | 0.400 ± 0.275 | 0.777 ± 0.707 | 0.762 ± 0.500 | 1.829 ± 2.367 | 0.302 ± 0.155 | 1.027 ± 0.211 | 0.748 ± 0.509 | 1.701 ± 2.148 | 0.330 ± 0.263 |
| ACOMB MFS | Event 1 | 0.170 ± 0.140 | 0.059 ± 0.054 | 0.070 ± 0.034 | 0.905 ± 0.047 | 0.000 ± 0.000 | 0.314 ± 0.089 | 0.288 ± 0.136 | 0.610 ± 0.184 | 0.500 ± 0.250 | 0.174 ± 0.150 | 0.164 ± 0.098 | 0.778 ± 0.133 |
| | Event 2 | 0.420 ± 0.140 | 0.149 ± 0.042 | 0.084 ± 0.016 | 0.889 ± 0.022 | 1.170 ± 1.010 | 0.288 ± 0.600 | 0.276 ± 0.071 | 0.636 ± 0.094 | 1.250 ± 0.430 | 0.068 ± 0.057 | 0.075 ± 0.050 | 0.901 ± 0.066 |
| | Event 3 | 0.830 ± 0.140 | 0.699 ± 0.157 | 1.924 ± 0.297 | 0.430 ± 0.088 | 0.830 ± 0.140 | 1.084 ± 0.144 | 3.337 ± 0.781 | 0.003 ± 0.231 | 0.750 ± 0.250 | 0.927 ± 0.032 | 2.427 ± 0.283 | 0.281 ± 0.084 |
| | **Average** | 0.473 ± 0.333 | 0.302 ± 0.346 | 0.693 ± 1.067 | 0.741 ± 0.270 | 0.667 ± 0.602 | 0.562 ± 0.452 | 1.300 ± 1.763 | 0.416 ± 0.358 | 0.833 ± 0.382 | 0.390 ± 0.468 | 0.889 ± 1.333 | 0.653 ± 0.328 |
| STOCKSFIELD | Event 1 | 1.420 ± 0.580 | 0.585 ± 0.115 | 1.015 ± 0.433 | 0.471 ± 0.226 | 1.330 ± 0.720 | 0.999 ± 0.096 | 2.862 ± 0.417 | -0.491 ± 0.217 | 1.250 ± 0.660 | 0.686 ± 0.030 | 1.497 ± 0.114 | 0.220 ± 0.060 |
| | Event 2 | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × |
| | Event 3 | 1.750 ± 0.000 | 0.964 ± 0.009 | 2.597 ± 0.062 | -0.512 ± 0.036 | 1.000 ± 0.660 | 1.009 ± 0.015 | 2.774 ± 0.088 | -0.615 ± 0.051 | 1.500 ± 0.430 | 0.916 ± 0.036 | 2.351 ± 0.140 | -0.369 ± 0.082 |
| | **Average** | 1.057 ± 0.930 | 0.516 ± 0.486 | 1.204 ± 1.309 | -0.014 ± 0.492 | 0.777 ± 0.693 | 0.669 ± 0.580 | 1.879 ± 1.628 | -0.369 ± 0.325 | 0.917 ± 0.804 | 0.534 ± 0.477 | 1.283 ± 1.190 | -0.050 ± 0.298 |
| NUNNYKIRK | Event 1 | 2.750 ± 3.910 | 0.248 ± 0.085 | 0.382 ± 0.192 | -0.646 ± 0.828 | 7.250 ± 0.000 | 0.263 ± 0.028 | 0.559 ± 0.165 | -1.414 ± 0.714 | 4.170 ± 3.740 | 0.315 ± 0.007 | 0.515 ± 0.217 | -1.219 ± 0.935 |
| | Event 2 | 1.920 ± 0.140 | 0.182 ± 0.053 | 0.330 ± 0.039 | 0.342 ± 0.079 | 1.920 ± 0.140 | 0.248 ± 0.115 | 0.418 ± 0.227 | 0.168 ± 0.451 | 2.000 ± 0.000 | 0.214 ± 0.110 | 0.336 ± 0.149 | 0.330 ± 0.298 |
| | Event 3 | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × |
| | **Average** | 1.557 ± 1.411 | 0.143 ± 0.128 | 0.237 ± 0.207 | -0.101 ± 0.502 | 3.057 ± 3.756 | 0.170 ± 0.148 | 0.326 ± 0.291 | -0.415 ± 0.869 | 2.057 ± 2.086 | 0.176 ± 0.161 | 0.284 ± 0.261 | -0.296 ± 0.816 |
| KNITSLEY | Event 1 | 0.330 ± 0.140 | 0.072 ± 0.061 | 0.025 ± 0.021 | 0.953 ± 0.040 | 0.330 ± 0.140 | 0.128 ± 0.085 | 0.076 ± 0.048 | 0.857 ± 0.089 | 0.170 ± 0.140 | 0.237 ± 0.044 | 0.190 ± 0.067 | 0.645 ± 0.124 |
| | Event 2 | 0.580 ± 0.140 | 0.186 ± 0.087 | 0.125 ± 0.107 | 0.892 ± 0.092 | 0.920 ± 0.760 | 0.395 ± 0.099 | 0.443 ± 0.173 | 0.619 ± 0.149 | 1.000 ± 0.660 | 0.345 ± 0.052 | 0.409 ± 0.189 | 0.648 ± 0.162 |
| | Event 3 | 1.250 ± 0.870 | 0.189 ± 0.022 | 0.257 ± 0.097 | -0.071 ± 0.404 | 0.330 ± 0.290 | 0.135 ± 0.023 | 0.092 ± 0.007 | 0.617 ± 0.030 | 0.920 ± 0.520 | 0.223 ± 0.034 | 0.213 ± 0.061 | 0.113 ± 0.256 |
| | **Average** | 0.720 ± 0.476 | 0.149 ± 0.067 | 0.136 ± 0.116 | 0.591 ± 0.574 | 0.527 ± 0.341 | 0.219 ± 0.152 | 0.204 ± 0.207 | 0.698 ± 0.138 | 0.697 ± 0.458 | 0.268 ± 0.067 | 0.271 ± 0.120 | 0.469 ± 0.308 |
| KIELDER | Event 1 | 0.330 ± 0.380 | 0.086 ± 0.051 | 0.040 ± 0.041 | 0.765 ± 0.242 | 0.420 ± 0.290 | 0.142 ± 0.021 | 0.066 ± 0.013 | 0.613 ± 0.076 | 0.080 ± 0.140 | 0.037 ± 0.014 | 0.012 ± 0.003 | 0.928 ± 0.018 |
| | Event 2 | 2.750 ± 0.000 | 0.064 ± 0.043 | 0.072 ± 0.008 | -0.364 ± 0.160 | 1.500 ± 1.250 | 0.089 ± 0.015 | 0.078 ± 0.033 | -0.481 ± 0.627 | 3.830 ± 2.770 | 0.068 ± 0.007 | 0.076 ± 0.021 | -0.411 ± 0.021 |
| | Event 3 | 2.330 ± 0.950 | 0.043 ± 0.022 | 0.025 ± 0.023 | 0.341 ± 0.593 | 7.750 ± 0.000 | 0.066 ± 0.007 | 0.083 ± 0.022 | -0.239 ± 0.333 | 1.750 ± 1.250 | 0.048 ± 0.030 | 0.024 ± 0.014 | 0.354 ± 0.376 |
| | **Average** | 1.803 ± 0.443 | 0.064 ± 0.039 | 0.046 ± 0.024 | 0.247 ± 0.332 | 3.223 ± 0.513 | 0.099 ± 0.014 | 0.076 ± 0.023 | -0.036 ± 0.345 | 1.887 ± 1.387 | 0.051 ± 0.017 | 0.037 ± 0.013 | 0.290 ± 0.138 |

Table 10: Catchment-level forecasting 4 hours before peak. Metrics are mean±SD across seeds. Errors: peak timing $\Delta t_{\text{peak}}$ (h)↓, peak height $\Delta h_{\text{peak}}$ (m)↓, MSE↓, NSE↑.

| Data | Split | APILANET | | | | CROSSFORMER | | | | TSMIXER | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $\Delta t_{\text{peak}}$↓ | $\Delta h_{\text{peak}}$↓ | MSE↓ | NSE↑ | $\Delta t_{\text{peak}}$↓ | $\Delta h_{\text{peak}}$↓ | MSE↓ | NSE↑ | $\Delta t_{\text{peak}}$↓ | $\Delta h_{\text{peak}}$↓ | MSE↓ | NSE↑ |
| ACOMB GRN | Event 1 | 0.580 ± 0.140 | 0.383 ± 0.039 | 0.291 ± 0.060 | 0.526 ± 0.098 | 0.170 ± 0.140 | 0.508 ± 0.076 | 0.505 ± 0.186 | 0.176 ± 0.304 | 0.250 ± 0.000 | 0.432 ± 0.056 | 0.314 ± 0.059 | 0.489 ± 0.097 |
| | Event 2 | 0.500 ± 0.250 | 0.382 ± 0.015 | 0.324 ± 0.057 | 0.074 ± 0.163 | 0.330 ± 0.380 | 0.424 ± 0.090 | 0.457 ± 0.303 | 0.305 ± 0.865 | 0.750 ± 0.500 | 0.307 ± 0.039 | 0.253 ± 0.098 | 0.275 ± 0.280 |
| | Event 3 | 2.830 ± 1.180 | 1.383 ± 0.070 | 6.145 ± 0.153 | -0.416 ± 0.036 | 3.750 ± 0.000 | 1.340 ± 0.102 | 6.276 ± 1.081 | -0.446 ± 0.249 | 1.750 ± 0.500 | 1.429 ± 0.093 | 5.549 ± 0.581 | -0.279 ± 0.134 |
| | **Average** | 1.303 ± 1.323 | 0.716 ± 0.578 | 2.253 ± 3.370 | 0.061 ± 0.471 | 1.417 ± 2.022 | 0.757 ± 0.506 | 2.413 ± 3.346 | 0.012 ± 0.402 | 0.917 ± 0.764 | 0.723 ± 0.615 | 2.039 ± 3.040 | 0.162 ± 0.396 |
| ACOMB MFS | Event 1 | 0.080 ± 0.140 | 0.123 ± 0.030 | 0.069 ± 0.021 | 0.863 ± 0.042 | 0.420 ± 0.140 | 0.208 ± 0.121 | 0.214 ± 0.145 | 0.576 ± 0.287 | 0.170 ± 0.140 | 0.195 ± 0.127 | 0.164 ± 0.152 | 0.676 ± 0.301 |
| | Event 2 | 0.500 ± 0.430 | 0.175 ± 0.082 | 0.147 ± 0.062 | 0.750 ± 0.105 | 0.670 ± 0.950 | 0.295 ± 0.037 | 0.279 ± 0.073 | 0.527 ± 0.124 | 1.670 ± 0.630 | 0.096 ± 0.082 | 0.115 ± 0.072 | 0.806 ± 0.0.121 |
| | Event 3 | 1.000 ± 0.250 | 0.732 ± 0.107 | 2.389 ± 0.149 | 0.109 ± 0.056 | 3.750 ± 0.000 | 1.126 ± 0.076 | 4.406 ± 0.505 | -0.642 ± 0.188 | 2.000 ± 1.520 | 1.072 ± 0.106 | 3.662 ± 0.684 | -0.365 ± 0.255 |
| | **Average** | 0.527 ± 0.461 | 0.343 ± 0.338 | 0.868 ± 1.318 | 0.574 ± 0.407 | 1.613 ± 1.855 | 0.543 ± 0.507 | 1.633 ± 2.402 | 0.154 ± 0.690 | 1.280 ± 0.975 | 0.454 ± 0.537 | 1.314 ± 2.034 | 0.372 ± 0.642 |
| STOCKSFIELD | Event 1 | 1.420 ± 0.760 | 0.588 ± 0.091 | 1.233 ± 0.569 | 0.191 ± 0.374 | 3.250 ± 0.660 | 1.032 ± 0.052 | 3.709 ± 0.157 | -1.433 ± 0.103 | 1.920 ± 1.040 | 0.724 ± 0.049 | 1.495 ± 0.260 | 0.019 ± 0.171 |
| | Event 2 | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × |
| | Event 3 | 2.420 ± 1.530 | 0.893 ± 0.085 | 2.994 ± 0.136 | -1.349 ± 0.106 | 2.750 ± 0.250 | 0.742 ± 0.022 | 2.006 ± 0.278 | -0.574 ± 0.218 | 3.500 ± 0.430 | 0.821 ± 0.080 | 2.571 ± 0.227 | -1.017 ± 0.178 |
| | **Average** | 1.280 ± 1.216 | 0.494 ± 0.454 | 1.409 ± 1.505 | -0.386 ± 0.839 | 2.000 ± 1.750 | 0.591 ± 0.532 | 1.905 ± 1.856 | -0.669 ± 0.722 | 1.807 ± 1.753 | 0.515 ± 0.449 | 1.355 ± 1.291 | -0.333 ± 0.593 |
| NUNNYKIRK | Event 1 | 0.750 ± 1.300 | 0.138 ± 0.096 | 0.239 ± 0.162 | -0.071 ± 0.724 | 4.000 ± 2.170 | 0.347 ± 0.008 | 1.011 ± 0.238 | -3.515 ± 1.064 | 1.170 ± 0.950 | 0.305 ± 0.098 | 0.695 ± 0.471 | -2.105 ± 2.103 |
| | Event 2 | 2.580 ± 0.760 | 0.116 ± 0.026 | 0.108 ± 0.044 | 0.513 ± 0.200 | 1.250 ± 0.250 | 0.248 ± 0.024 | 0.319 ± 0.061 | -0.442 ± 0.277 | 2.080 ± 0.800 | 0.132 ± 0.046 | 0.129 ± 0.078 | 0.419 ± 0.349 |
| | Event 3 | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × |
| | **Average** | 1.110 ± 1.327 | 0.085 ± 0.074 | 0.116 ± 0.120 | 0.147 ± 0.319 | 1.750 ± 2.046 | 0.198 ± 0.179 | 0.443 ± 0.517 | -1.319 ± 1.915 | 1.083 ± 1.043 | 0.146 ± 0.153 | 0.275 ± 0.370 | -0.562 ± 1.352 |
| KNITSLEY | Event 1 | 0.580 ± 0.580 | 0.054 ± 0.046 | 0.056 ± 0.033 | 0.843 ± 0.092 | 0.330 ± 0.140 | 0.107 ± 0.099 | 0.074 ± 0.072 | 0.792 ± 0.203 | 0.000 ± 0.000 | 0.237 ± 0.060 | 0.205 ± 0.123 | 0.425 ± 0.345 |
| | Event 2 | 0.420 ± 0.520 | 0.100 ± 0.055 | 0.059 ± 0.025 | 0.923 ± 0.032 | 0.330 ± 0.380 | 0.359 ± 0.118 | 0.302 ± 0.179 | 0.609 ± 0.232 | 0.580 ± 0.290 | 0.215 ± 0.130 | 0.206 ± 0.087 | 0.733 ± 0.112 |
| | Event 3 | 1.080 ± 1.460 | 0.080 ± 0.013 | 0.137 ± 0.089 | 0.214 ± 0.510 | 3.750 ± 0.000 | 0.085 ± 0.015 | 0.129 ± 0.068 | 0.262 ± 0.393 | 2.500 ± 2.170 | 0.090 ± 0.032 | 0.058 ± 0.021 | 0.667 ± 0.122 |
| | **Average** | 0.693 ± 0.344 | 0.078 ± 0.023 | 0.084 ± 0.046 | 0.660 ± 0.388 | 1.750 ± 1.975 | 0.184 ± 0.153 | 0.168 ± 0.119 | 0.554 ± 0.269 | 1.027 ± 1.308 | 0.181 ± 0.079 | 0.156 ± 0.085 | 0.608 ± 0.162 |
| KIELDER | Event 1 | 0.170 ± 0.140 | 0.030 ± 0.022 | 0.009 ± 0.006 | 0.936 ± 0.040 | 0.330 ± 0.380 | 0.055 ± 0.030 | 0.023 ± 0.010 | 0.846 ± 0.069 | 0.420 ± 0.380 | 0.064 ± 0.041 | 0.026 ± 0.018 | 0.824 ± 0.121 |
| | Event 2 | 4.170 ± 0.520 | 0.049 ± 0.025 | 0.033 ± 0.008 | -1.066 ± 0.513 | 2.000 ± 2.380 | 0.098 ± 0.009 | 0.074 ± 0.011 | -3.584 ± 0.683 | 1.580 ± 1.700 | 0.053 ± 0.027 | 0.035 ± 0.035 | -1.161 ± 2.177 |
| | Event 3 | 2.000 ± 0.430 | 0.056 ± 0.017 | 0.021 ± 0.011 | -2.070 ± 1.639 | 1.500 ± 1.500 | 0.067 ± 0.023 | 0.027 ± 0.019 | -2.968 ± 2.826 | 0.750 ± 0.660 | 0.037 ± 0.029 | 0.012 ± 0.007 | -0.794 ± 0.946 |
| | **Average** | 2.113 ± 0.363 | 0.045 ± 0.021 | 0.021 ± 0.008 | -0.733 ± 0.731 | 1.277 ± 1.420 | 0.073 ± 0.021 | 0.041 ± 0.013 | -1.902 ± 1.193 | 0.917 ± 0.913 | 0.051 ± 0.032 | 0.024 ± 0.020 | -0.377 ± 1.081 |

Table 11: Catchment-level forecasting 2 hours before peak. Metrics are mean±SD across seeds. Errors: peak timing $\Delta t_{\text{peak}}$ (h)↓, peak height $\Delta h_{\text{peak}}$ (m)↓, MSE↓, NSE↑.

| Data | Split | APILANET | | | | CROSSFORMER | | | | TSMIXER | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $\Delta t_{\text{peak}}$↓ | $\Delta h_{\text{peak}}$↓ | MSE↓ | NSE↑ | $\Delta t_{\text{peak}}$↓ | $\Delta h_{\text{peak}}$↓ | MSE↓ | NSE↑ | $\Delta t_{\text{peak}}$↓ | $\Delta h_{\text{peak}}$↓ | MSE↓ | NSE↑ |
| ACOMB GRN | Event 1 | 0.250 ± 0.250 | 0.403 ± 0.057 | 0.281 ± 0.062 | 0.444 ± 0.123 | 0.080 ± 0.140 | 0.506 ± 0.080 | 0.557 ± 0.259 | -0.102 ± 0.514 | 0.670 ± 0.380 | 0.497 ± 0.016 | 0.485 ± 0.101 | 0.041 ± 0.200 |
| | Event 2 | 0.330 ± 0.140 | 0.305 ± 0.052 | 0.161 ± 0.082 | 0.551 ± 0.230 | 0.580 ± 0.380 | 0.360 ± 0.061 | 0.211 ± 0.124 | 0.412 ± 0.346 | 0.580 ± 0.380 | 0.364 ± 0.029 | 0.209 ± 0.025 | 0.415 ± 0.070 |
| | Event 3 | 5.33 ± 0.380 | 1.267 ± 0.113 | 6.142 ± 0.967 | -1.196 ± 0.346 | 3.420 ± 2.040 | 1.427 ± 0.049 | 6.110 ± 0.622 | -1.183 ± 0.222 | 2.500 ± 2.180 | 1.298 ± 0.167 | 5.001 ± 1.337 | -0.790 ± 0.478 |
| | **Average** | 1.970 ± 2.910 | 0.658 ± 0.529 | 2.195 ± 3.419 | -0.067 ± 0.979 | 1.360 ± 1.801 | 0.764 ± 0.579 | 2.293 ± 3.310 | -0.291 ± 0.814 | 1.250 ± 1.083 | 0.720 ± 0.505 | 1.898 ± 2.691 | -0.111 ± 0.617 |
| ACOMB MFS | Event 1 | 0.500 ± 0.250 | 0.178 ± 0.041 | 0.192 ± 0.139 | 0.522 ± 0.347 | 0.250 ± 0.000 | 0.243 ± 0.101 | 0.306 ± 0.279 | 0.253 ± 0.680 | 0.330 ± 0.140 | 0.172 ± 0.026 | 0.107 ± 0.021 | 0.738 ± 0.051 |
| | Event 2 | 0.750 ± 0.250 | 0.051 ± 0.030 | 0.031 ± 0.017 | 0.885 ± 0.063 | 1.500 ± 0.250 | 0.206 ± 0.024 | 0.166 ± 0.009 | 0.391 ± 0.034 | 0.750 ± 0.250 | 0.147 ± 0.083 | 0.109 ± 0.082 | 0.601 ± 0.299 |
| | Event 3 | 4.170 ± 2.320 | 0.737 ± 0.064 | 3.258 ± 0.925 | -1.162 ± 0.614 | 5.750 ± 0.000 | 1.021 ± 0.034 | 4.988 ± 0.332 | -2.310 ± 0.220 | 4.330 ± 2.450 | 0.972 ± 0.031 | 3.349 ± 0.202 | -1.222 ± 0.134 |
| | **Average** | 1.807 ± 2.051 | 0.322 ± 0.365 | 1.160 ± 1.819 | 0.082 ± 1.092 | 2.500 ± 2.883 | 0.490 ± 0.460 | 1.820 ± 2.744 | -0.555 ± 1.521 | 1.803 ± 2.198 | 0.430 ± 0.469 | 1.188 ± 1.873 | 0.039 ± 1.095 |
| STOCKSFIELD | Event 1 | 1.330 ± 0.140 | 0.679 ± 0.104 | 1.593 ± 0.552 | -0.634 ± 0.566 | 1.330 ± 0.380 | 0.997 ± 0.056 | 3.189 ± 0.435 | -2.270 ± 0.446 | 1.080 ± 0.380 | 0.721 ± 0.097 | 1.586 ± 0.460 | -0.627 ± 0.472 |
| | Event 2 | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × |
| | Event 3 | 5.670 ± 0.140 | 0.753 ± 0.052 | 1.990 ± 0.099 | -2.614 ± 0.181 | 5.330 ± 0.380 | 0.788 ± 0.069 | 2.004 ± 0.386 | -2.639 ± 0.701 | 5.670 ± 0.140 | 0.728 ± 0.006 | 1.843 ± 0.258 | -2.347 ± 0.469 |
| | **Average** | 2.333 ± 2.965 | 0.477 ± 0.415 | 1.194 ± 1.053 | -1.083 ± 1.364 | 2.220 ± 2.774 | 0.595 ± 0.526 | 1.731 ± 1.612 | -1.636 ± 1.429 | 2.250 ± 3.011 | 0.483 ± 0.418 | 1.143 ± 0.998 | -0.991 ± 1.215 |
| NUNNYKIRK | Event 1 | 0.670 ± 0.140 | 0.156 ± 0.112 | 0.459 ± 0.240 | -4.731 ± 2.999 | 1.330 ± 0.760 | 0.362 ± 0.023 | 1.146 ± 0.093 | -13.312 ± 1.164 | 0.830 ± 0.720 | 0.334 ± 0.061 | 0.773 ± 0.361 | -8.646 ± 4.509 |
| | Event 2 | 1.750 ± 0.250 | 0.049 ± 0.040 | 0.052 ± 0.029 | 0.238 ± 0.427 | 1.000 ± 0.000 | 0.096 ± 0.067 | 0.069 ± 0.068 | -0.009 ± 0.994 | 1.830 ± 0.800 | 0.084 ± 0.037 | 0.074 ± 0.032 | -0.081 ± 0.482 |
| | Event 3 | msd×× | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × | × ± × |
| | **Average** | 0.807 ± 0.883 | 0.068 ± 0.080 | 0.170 ± 0.251 | -1.498 ± 2.803 | 0.777 ± 0.693 | 0.153 ± 0.188 | 0.405 ± 0.643 | -4.440 ± 7.683 | 0.887 ± 0.916 | 0.139 ± 0.174 | 0.282 ± 0.427 | -2.909 ± 4.969 |
| KNITSLEY | Event 1 | 0.330 ± 0.290 | 0.027 ± 0.023 | 0.045 ± 0.030 | 0.724 ± 0.186 | 0.420 ± 0.140 | 0.146 ± 0.049 | 0.106 ± 0.009 | 0.355 ± 0.055 | 0.170 ± 0.140 | 0.210 ± 0.035 | 0.242 ± 0.009 | -0.477 ± 0.058 |
| | Event 2 | 0.000 ± 0.000 | 0.078 ± 0.054 | 0.068 ± 0.036 | 0.885 ± 0.061 | 0.000 ± 0.000 | 0.084 ± 0.008 | 0.108 ± 0.029 | 0.818 ± 0.050 | 0.330 ± 0.380 | 0.148 ± 0.096 | 0.256 ± 0.111 | 0.567 ± 0.187 |
| | Event 3 | 5.330 ± 0.140 | 0.097 ± 0.056 | 0.123 ± 0.032 | -1.641 ± 0.689 | 1.920 ± 3.320 | 0.023 ± 0.010 | 0.036 ± 0.009 | 0.225 ± 0.212 | 4.080 ± 2.890 | 0.004 ± 0.004 | 0.057 ± 0.013 | -0.237 ± 0.272 |
| | **Average** | 1.887 ± 2.987 | 0.067 ± 0.036 | 0.079 ± 0.040 | -0.011 ± 1.414 | 0.780 ± 1.009 | 0.084 ± 0.062 | 0.083 ± 0.041 | 0.466 ± 0.312 | 1.527 ± 2.213 | 0.121 ± 0.106 | 0.185 ± 0.111 | -0.049 ± 0.547 |
| KIELDER | Event 1 | 0.580 ± 0.250 | 0.045 ± 0.044 | 0.016 ± 0.012 | 0.829 ± 0.138 | 0.420 ± 0.140 | 0.034 ± 0.023 | 0.018 ± 0.008 | 0.809 ± 0.094 | 0.420 ± 0.290 | 0.114 ± 0.030 | 0.061 ± 0.025 | 0.342 ± 0.267 |
| | Event 2 | 1.080 ± 0.080 | 0.037 ± 0.007 | 0.011 ± 0.006 | -0.107 ± 0.270 | 1.670 ± 0.380 | 0.030 ± 0.012 | 0.090 ± 0.045 | -0.591 ± 0.792 | 1.250 ± 0.500 | 0.030 ± 0.003 | 0.063 ± 0.015 | -0.857 ± 1.035 |
| | Event 3 | 0.170 ± 0.140 | 0.009 ± 0.010 | 0.011 ± 0.008 | 0.527 ± 0.366 | 0.580 ± 0.380 | 0.025 ± 0.024 | 0.240 ± 0.016 | -0.053 ± 0.753 | 1.830 ± 2.550 | 0.022 ± 0.007 | 0.420 ± 0.028 | -0.911 ± 1.255 |
| | **Average** | 0.610 ± 0.157 | 0.028 ± 0.020 | 0.013 ± 0.009 | 0.416 ± 0.258 | 0.890 ± 0.300 | 0.030 ± 0.020 | 0.116 ± 0.023 | 0.055 ± 0.546 | 1.167 ± 1.113 | 0.055 ± 0.013 | 0.181 ± 0.023 | -0.475 ± 0.852 |