WHAT DOES A THERMAL SURROGATE LOSE WITHOUT THE FLOW FIELD? FORECASTING VALVE-INDUCED TRANSIENTS IN LIQUID-COOLED DATACENTERS WITH GRAPH NEURAL ODES

Anonymous authors
Paper under double-blind review

ABSTRACT

Learned surrogates of thermal-fluid plants are usually trained and evaluated with the flow field given. In a liquid-cooled datacenter, closing one cooling channel's valve pushes its flow onto the sibling channels within seconds, and a deployed surrogate rarely observes that post-event flow. We measure what this costs and whether the missing flow can be supplied. On a transient simulator of the Frontier supercomputer's cooling plant, we train graph surrogates under four information regimes: true flow, flow frozen at the forecast start, a closed-form branch-conductance estimate computed from pre-event measurements and the valve command, and frozen flow plus the command given to the network. We evaluate each on 49 held-out closures. Without post-event flow, the sibling-channel error of a flow-structured graph neural ODE over the first minute rises about 32-fold (from 0.032 to 1.013 K), two baselines fail in the same way, and every model predicts the wrong direction of change on the closed channel in at least 47 of the 49 events. Event-locked controls show that the gap appears only at the event. The closed-form estimate, a standard virtual flow meter, recovers 99% of the gap. Given the command instead, the network recovers 85% on the closed channel but only 6% on its siblings: it learns what a valve does to its own channel, not how the flow redistributes to its neighbours.

1 INTRODUCTION

Learned surrogate of a liquid-cooled datacenter can score well on held-out data and still get every valve event wrong. We trained a graph neural ODE on simulated transients of the Frontier supercomputer's cooling plant, giving it the coolant flow measured when each forecast starts. On event-free test rollouts, its 60 s forecasts reach a skill of 0.71 against persistence. Still, after each of the 49 held-out valve closures it predicts the wrong direction of change for the channel whose valve closed, that channel heats by a median of 4.3 K within a minute, and the model forecasts a slight cooling (Figure 1b).

The reason for this is physical. Closing one cooling channel's valve pushes most of its flow onto the sibling channels in the same cabinet (Figure 1a); in the plant we study that they absorb 69–92% of the flow the closed channel loses. Flow settles within seconds while temperatures take minutes to follow, so over a forecast horizon the flow field drives what happens next. A surrogate that sees the flow field from before the event sees no event at all. This is the usual situation in deployment, but the flow is rarely measured on every branch, and controllers that move valves, such as the reinforcement-learning agents of Naug et al. (2025), change the very flow field the surrogate relies on.

These observations lead us to two questions. **Q1:** when an actuator redistributes flow that is then not observed, how much forecast accuracy does a learned temperature surrogate lose, where in the plant, and does the loss depend on the architecture? **Q2:** can the missing flow be supplied without measuring it, and which recovers more: a closed-form hydraulic estimate computed from pre-event measurements and the valve command, or the command itself given to the network? To

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

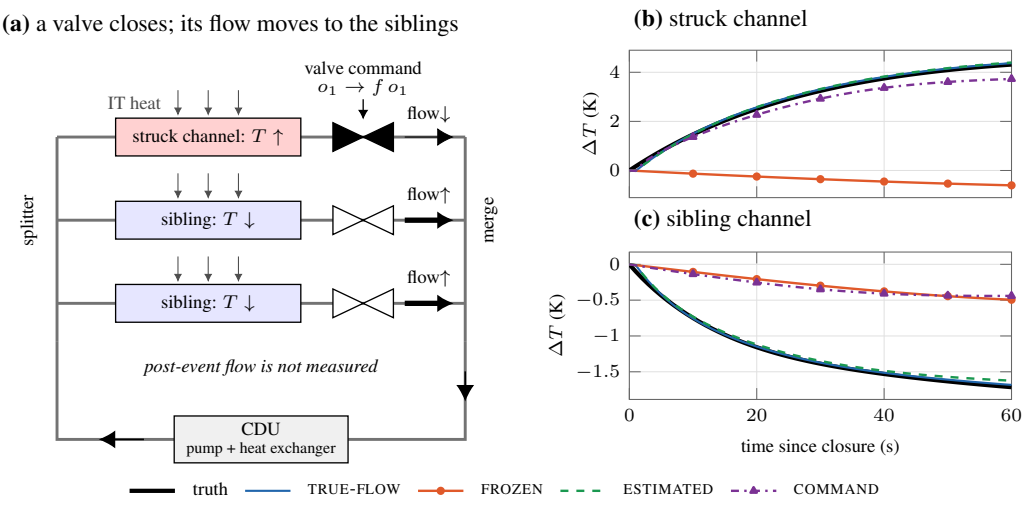


Figure 1: **Told only the flow from before a valve closure, a surrogate forecasts the wrong direction; a closed-form flow estimate restores the forecast.** (a) Closing the valve of one cooling channel pushes most of its flow onto the two sibling channels of the same cabinet (69–92% over our training closures): the struck channel heats and the siblings cool (median changes $+4.3$ and -1.5 K over the first 60 s). (b, c) One test closure, the median of the 49 by the struck channel's true temperature rise over 60 s: temperature change of the struck channel and of the sibling with the larger change, truth and AdvGNODE forecasts (seed 0) in four regimes, each trained in its own regime: TRUE-FLOW (the true flow), FROZEN (flow frozen at the forecast start), ESTIMATED (frozen flow plus a closed-form estimate of the redistribution) and COMMAND (frozen flow plus the valve command). Under FROZEN the struck-channel forecast has the wrong sign, as in all 49 test closures.

our knowledge neither has been answered. Learned thermal-fluid surrogates take flow as a measured or simulated input, compute it in a separate module, or predict it jointly (Almukhametov et al., 2026; Boca de Giuli et al., 2024; Boghetti et al., 2026; Bott et al., 2025), and virtual flow metering (Song et al., 2013; Bikmukhametov & Jäschke, 2020) has not been evaluated as the input to a learned forecaster.

We answer both questions on a transient model of the Frontier plant from the ExaDigiT digital twin (Brewer et al., 2024). Surrogates are trained under four information regimes: TRUE-FLOW (the true flow), FROZEN (flow frozen at the forecast start), ESTIMATED (frozen flow plus a closed-form estimate of the redistribution) and COMMAND (frozen flow plus the valve command). Each model is trained and evaluated in its own regime, so a gap between regimes actually measures the value of the information, not a train-test mismatch. We compare regimes by paired tests over 49 test closures, check that the differences are caused by the event, and then repeat the study for AdvGNODE, a graph neural ODE whose messages are scaled by flow, for the same network with flow as an ordinary input, and for a spatio-temporal graph convolutional network.

Without post-event flow, AdvGNODE's error on the sibling channels over the first minute rises about 32-fold, from 0.032 to 1.013 K, and the other two architectures fail in the same way. The closed-form estimate recovers 99% of this gap. Moreover, the command does not: given it, AdvGNODE recovers 85% of the gap on the closed channel but only 6% on its siblings.

Furthermore, part of this is expected, but four findings are not expected. First, the failure is qualitative. Under FROZEN every model predicts the wrong sign on the closed channel in at least 47 of the 49 closures and only 27–31% of the sibling change, while scoring well on event-free data. Second, it comes from the missing information, not from training. The TRUE-FLOW model given frozen flow fails just as the FROZEN model does. Third, the valve command is not enough. The network learns what a valve does to its own channel but not what it does to the neighbours, whose flow changes through a pressure effect that flow-following messages do not carry (Section 7). Fourth, a textbook estimator identified from training data closes nearly the whole gap.

Our contributions are: (i) The measurement of the forecast cost of unobserved post-actuation flow for learned thermal-fluid surrogates, localised by component, repeated across three architectures and attributed by controls to the event and to the missing information. (ii) A controlled comparison of four ways of supplying post-event flow, in which a standard branch-conductance estimate recovers what the raw valve command does not. (iii) An event-anchored evaluation protocol on a graph derived from the simulator's connectivity and checked against its mass balance, with code to be released after acceptance.

2 Related Work

Graph networks and neural ODEs are now standard tools for emulating physical simulators (Pfaff et al., 2021; Brandstetter et al., 2022; Chen et al., 2018), and advection-based continuous-time models have been used successfully for weather forecasting (Verma et al., 2024). For thermal-fluid networks specifically, the closest work is a graph neural ODE for reactor thermal-hydraulics whose messages are scaled by loop mass flow (Almukhametov et al., 2026). There the loop flow is a measured input, and partial observability refers to missing temperature sensors, not missing flow. District heating surrogates make the same kind of assumption in different ways: flows are supplied by a simulator (Boca de Giuli et al., 2024), computed by a learned hydraulic module that feeds a physical thermal model (Boghetti et al., 2026), predicted jointly in a steady-state setting (Bott et al., 2025), or reduced to the plant's total flow apportioned to consumers by peak demand, at a 20-minute resolution where hydraulic transients are treated as instantaneous (Boussaid et al., 2024). In every case the flow field after an actuation is either given or never isolated. None of these works measures what a temperature forecaster loses when that flow field is not observed.

The plant we study comes from the ExaDigiT digital twin of the Frontier supercomputer (Brewer et al., 2024), whose transient thermo-fluid models are built in Modelica (Kumar et al., 2024; Greenwood et al., 2024). The same twin underlies LC-Opt, a reinforcement-learning benchmark in which cabinet-level valve positions are control actions (Naug et al., 2025); such valve moves are exactly the events a surrogate would have to forecast through. A pre-surrogate analysis of three other ExaDigiT plants found negligible thermal coupling between CDUs and recommended independent per-CDU models (Nigatu et al., 2026). Their plants have no channel-level valve actuation among the inputs; the coupling we study lies inside a single cabinet, where closing one channel pushes flow onto its siblings. Learned models trained on Frontier telemetry take measured loop flows as inputs (Jadhav & Liu, 2026).

Computing a branch flow from a valve's command, its characteristic curve and a pressure difference is established practice in building services (Song et al., 2013; Kim et al., 2025) and in oil and gas production (Bikmukhametov & Jäschke, 2020). The estimator we use belongs to this family, and we do not claim it as new. What has not been studied is how well such an estimate serves as the input to a learned temperature forecaster, and whether it is needed at all when the network could instead be given the valve command and left to learn the redistribution itself.

Thermal models of hydronic systems often treat parallel branches as independent; accounting for their hydraulic interaction reduced zone-temperature error by about 11% in a recent grey-box study (Shakhesi et al., 2026). In time-series forecasting, exogenous covariates are modelled explicitly (Wang et al., 2024), and forecasters that rely on them degrade by 32–62% when those covariates are corrupted or missing (Hu & Ai, 2026). There the missing channel is treated as generic noise. In our setting it is physically determined by a known actuation, which is what makes it possible to reconstruct.

3 Forecasting Through an Actuation Event

We use the transient thermo-fluid model of the Frontier liquid-cooling plant from the ExaDigiT digital twin (Brewer et al., 2024; Kumar et al., 2024), exported as a Functional Mock-up Unit and distributed with LC-Opt (Naug et al., 2025). Five compute blocks are connected in parallel to a shared hot-water loop. In each block, a coolant distribution unit (CDU) pumps a secondary loop through a cabinet that splits into three parallel cooling channels; each channel has its own valve, flow resistance and cold plate, and the three branches are not identical (the third carries an extra

resistance). IT heat load follows a recorded workload profile and the outdoor wet-bulb temperature follows weather data, so the plant is never in a true steady state.

We derive the graph from the simulator's port connectivity rather than from component names. Nodes are components with their own fluid-temperature state. A chain of stateless components between two such nodes, such as a valve followed by a resistance, becomes one *stream* edge whose mass flow $\dot{m}_e$ is read from the simulator; components that exchange heat without exchanging fluid are joined by *thermal* edges. The resulting graph has 185 nodes and 210 edges. We checked it by closing the mass balance at every volume against that volume's own accumulation term, which agrees to round-off. The surrogates use 178 nodes and 203 edges, after removing seven boundary nodes that impose pressure (Appendix A).

A closure event moves the valve of one channel from its base opening $v_0 \sim \mathcal{U}[0.3, 1.0]$ to fv_0, with $f \sim \mathcal{U}[0.10, 0.50]$, at a random time between 320 and 380 s of a 900 s rollout. Below $f = 0.10$ the struck channel becomes a nearly stagnant, heated dead end, which is a different regime from the one we study. Logging is at 1 s and the valve ramps in 1 s, so the event acts as a step; we write t_k for the last sample before it. Flow redistributes within about 2 s, while temperatures take roughly 500 s to settle. The two sibling channels absorb 69 to 92% of the flow the struck channel loses (mean 81% over training closures), and the fraction depends on the base openings, so it is not a constant of the plant. As a contrast class we also generate *setpoint* events, which step the CDU pressure setpoint: they change a cabinet's total flow but leave its split between channels unchanged. The remaining rollouts contain no event.

Let t_0 be the time at which a forecast starts. The surrogate receives the temperatures at t_0, the heat inputs, and one flow input $u_e(t)$ per stream edge for $t > t_0$. We compare four ways of filling $u_e(t)$ when an event falls inside the forecast window:

$$\text{TRUE-FLOW} : u_e(t) = \dot{m}_e(t), \qquad \text{FROZEN} : u_e(t) = \dot{m}_e(t_0),$$
$$\text{ESTIMATED} : u_e(t) = \dot{m}_e(t_0) + \mathbb{1}[t > t_k]\,\widehat{\Delta \dot{m}}_e, \quad \text{COMMAND} : u_e(t) = \dot{m}_e(t_0),\ \text{input } o_c(t),$$

where $o_c(t)$ is the commanded opening of the valve on channel c. TRUE-FLOW requires a flow measurement on every branch, continuously. FROZEN is what a surrogate sees if nothing tells it that the event happened. ESTIMATED and COMMAND use only information available at t_k: the pre-event state and the valve change that was commanded. They differ in who computes the redistribution: the closed-form estimator of Section 4.1, or the network itself. Every surrogate is trained and evaluated in the same regime, so a gap between regimes measures the value of the information rather than a mismatch between training and deployment.

4 Supplying the Missing Flow

4.1 A closed-form redistribution estimator

Each channel of a cabinet is a series of a linear valve and fixed resistances, so its flow is $\dot{m}_c = k_c(o_c)\,\Delta p$, with a conductance k_c that depends on the valve opening o_c. We identify the branch parameters from logged pre-event data; every branch fit has $R^2 > 0.9999$, and the identified valve coefficient equals the simulator's nominal value. Seen from the cabinet's two terminals, the rest of the plant acts as a source with an internal resistance, $\Delta p = p_s - R_{\text{ext}} \sum_c \dot{m}_c$. We identify R_{ext} per block from the 392 training closures, comparing each event's pre-event state with the window $[t_k + 20, t_k + 30]$ s. It is therefore an effective value that already includes the slow response of the CDU pump controller.

At forecast time the estimator uses only pre-event measurements and the commanded change: the pre-event flows and openings fix p_s, the post-event openings give new conductances, and solving the resulting linear system gives the post-event flow on every edge of the struck cabinet, which is then carried into the CDU loop of the same block. On the 49 validation closures, the median relative error of the estimated change in sibling flow is 1.3% (90th percentile 2.5%). The estimate is a step at $t_k + 1$. This is standard virtual flow metering (Section 2); what we test is its value as the input to a learned forecaster.

4.2 THE VALVE COMMAND AS AN INPUT

In the COMMAND regime the network receives the FROZEN flows and, as an extra input on each channel node, that channel's valve opening, which changes at $t_k + 1$. Neural differential equations driven by exogenous inputs are well established (Kidger et al., 2020); here the input is local, since each opening enters only at its own channel's node. This adds 64 parameters. The network now has the same inputs as the estimator: the pre-event flows, the old and new openings, and a training set from which the hydraulics could be learned. What it lacks is only the explicit hydraulic model.

4.3 SURROGATES

AdvGNODE: an advective graph neural ODE. AdvGNODE is not a new architecture but a graph neural ODE (Chen et al., 2018; Poli et al., 2019) whose messages are weighted by mass flow, as in Almukhametov et al. (2026); we name it only to refer to it. The state is the vector of node temperatures $\boldsymbol{T}(t)$, and the model learns $\mathrm{d}\boldsymbol{T}/\mathrm{d}t = F_\theta(\boldsymbol{T}, \boldsymbol{Q}, \boldsymbol{u})$, with F_θ a message-passing network (Gilmer et al., 2017) on the plant graph. Each node's temperature and heat input are encoded into $\boldsymbol{h}_i \in \mathbb{R}^{64}$. On a stream edge e the message is sent downstream under the current sign of the flow, and its size is set by the flow,

$$\boldsymbol{m}_e = w_e \cdot \mathrm{MLP}_{\mathrm{adv}}\big([\boldsymbol{h}_{\mathrm{up}}, \boldsymbol{h}_{\mathrm{down}}, \mathrm{type}_e]\big), \qquad w_e = |u_e|/s_{\mathrm{type}(e)},$$

where s is the training-set mean flow for that edge type, so an edge without flow carries no message. Thermal edges carry $\boldsymbol{m}_e = \mathrm{UA}_e\,\mathrm{MLP}_{\mathrm{th}}([\boldsymbol{h}_i, \boldsymbol{h}_j])$ with $\mathrm{UA}_e = \mathrm{softplus}(\theta_e)(w_e{+}\epsilon)^\alpha$ and a learned $\alpha \in [0.4, 1]$. A node update combines $\boldsymbol{h}_i$ with the summed messages, and a decoder returns $\mathrm{d}T_i/\mathrm{d}t$. We integrate with fourth-order Runge–Kutta at a 1 s step with two substeps, holding inputs fixed within each step. The model has 72,559 parameters.

Base is the same model with flow entering as an ordinary input feature to $\mathrm{MLP}_{\mathrm{adv}}$ instead of scaling its output, and unweighted thermal messages (72,577 parameters). It receives exactly the same information, so AdvGNODE versus base isolates the flow structure. *STGCN* is a spatio-temporal graph convolutional network adapted from Yu et al. (2018) (37,569 parameters); it receives flow as the summed flow weight of each node's incoming edges, and it is the only model that reads a history: its input is the state at t_0 and at the two samples before it. *Persistence* holds $\boldsymbol{T}(t_0)$.

5 EXPERIMENTAL PROTOCOL

We simulate 700 rollouts of 900 s at 1 s resolution: 490 with a closure, 105 with a setpoint event and 105 without an event. Rollouts are split 560/70/70 into training, validation and test sets, stratified by type; the test set contains 49 closures, 10 setpoint events and 11 event-free rollouts. Forecast windows start no earlier than 200 s, after the plant has relaxed from its initial condition.

All learned models share one schedule: 30 epochs of one-step teacher forcing, then 70 epochs of backpropagation through rollouts whose length doubles from 2 to 32 steps every 14 epochs. We use Adam (Kingma & Ba, 2015) with learning rate 10^{-3} and cosine decay, batches of 16 windows, gradient clipping at 1, and mean squared error on normalized temperature. Each epoch draws 12,000 windows, a quarter of them containing an event; without this, only 4% of windows would. The checkpoint is chosen by validation skill at 60 s on a fixed set of 300 validation event windows, averaged over three consecutive epochs. Training is deterministic on the GPU. AdvGNODE and STGCN are trained with three seeds in the TRUE-FLOW, FROZEN and (for AdvGNODE) ESTIMATED regimes, COMMAND with two seeds, and base with two seeds in both.

Test windows start at the event, $t_0 = t_k$, so the event falls on the first step, and are scored over 30 and 60 s. We report RMSE over the first h seconds, for $h \in \{10, 30, 60\}$, separately on the struck channel, the two sibling channels, the rest of the struck cabinet, the other four blocks, and all nodes; skill is $1 - \mathrm{RMSE}/\mathrm{RMSE}_{\mathrm{persistence}}$, the root-mean-square form of the skill score against a reference forecast (Murphy, 1988). Regimes are compared by paired tests over the 49 test closures (t-test and Wilcoxon signed-rank), averaging each event's error over seeds. The *recovery* of a regime X is the fraction of the gap between FROZEN and TRUE-FLOW that it closes, $R_X = \sum_e(\epsilon_e^{\text{FROZEN}} - \epsilon_e^X)/\sum_e(\epsilon_e^{\text{FROZEN}} - \epsilon_e^{\text{TRUE-FLOW}})$, where ϵ_e is the RMSE of test closure e averaged over the seeds the three regimes share. To check that differences are caused by the event, we also start windows 10,

Table 1: **Without post-event flow every architecture fails on the channels the closure acts on; the closed-form estimate restores near-TRUE-FLOW accuracy.** RMSE (K) over the first 60 s after the closure, pooled over the 49 test closures, their nodes and seconds; seed mean ± standard deviation over seeds ($n-1$ denominator). Lower is better; bold marks the lowest value in each column. Among the regimes that do not measure post-event flow (FROZEN, ESTIMATED, COMMAND), ESTIMATED is lowest in every column. STGCN's lower struck-channel error under TRUE-FLOW is not significant (paired $p = 0.41$). Appendix C gives all node sets and seed ranges.

Model	Regime	Seeds	Struck channel	Sibling channels	All nodes
Persistence	–	–	3.7299	1.3149	0.4343
AdvGNODE	TRUE-FLOW	3	0.0851 ± 0.0041	**0.0319** ± 0.0026	**0.0669** ± 0.0021
AdvGNODE	FROZEN	3	4.0209 ± 0.0007	1.0127 ± 0.0003	0.3375 ± 0.0005
AdvGNODE	ESTIMATED	3	0.0882 ± 0.0055	0.0421 ± 0.0015	0.0705 ± 0.0025
AdvGNODE	COMMAND	2	0.8589 ± 0.0577	0.9719 ± 0.0029	0.1580 ± 0.0076
Base	TRUE-FLOW	2	0.1205 ± 0.0056	0.1339 ± 0.0221	0.1034 ± 0.0163
Base	FROZEN	2	4.0146 ± 0.0054	1.0132 ± 0.0026	0.3433 ± 0.0028
STGCN	TRUE-FLOW	3	**0.0731** ± 0.0057	0.0917 ± 0.0150	0.0706 ± 0.0038
STGCN	FROZEN	3	3.9922 ± 0.0053	1.0425 ± 0.0032	0.3344 ± 0.0011

30, 60 and 110 s *before* it and compare errors before and after the event inside the same rollout; setpoint events, which do not redistribute flow between channels, serve as a contrast class. The primary comparison (sibling-channel error, TRUE-FLOW versus FROZEN versus ESTIMATED), its significance and effect-size thresholds, and the interpretation of the COMMAND regime were fixed before any test result was seen.

6 RESULTS

6.1 WHAT IS LOST WITHOUT POST-EVENT FLOW

Without post-event flow the forecasts collapse exactly where the event acts (Table 1). On the sibling channels, AdvGNODE's error rises from 0.032 K under TRUE-FLOW to 1.013 K under FROZEN, about 32 times higher; on the struck channel it rises from 0.085 to 4.02 K. The paired difference on the siblings is -0.89 K over 49 events ($p = 1.6 \times 10^{-19}$, t-test; $p = 3.6 \times 10^{-15}$, Wilcoxon), and every one of the 49 events shows it (Appendix C). The loss is nearly the same for all three architectures: under FROZEN the sibling error is 1.013, 1.013 and 1.043 K for AdvGNODE, base and STGCN.

The error alone understates what goes wrong. Table 2 asks whether each model predicts the right *direction* and *size* of the temperature change over 60 s. Under FROZEN, AdvGNODE predicts the wrong direction on the struck channel in all 49 events: the channel heats by a median of 4.3 K, and the model predicts a small cooling. Its median predicted change is about -0.46 K on the struck channel and on the siblings alike: it forecasts the same mild cooling everywhere and does not distinguish the channel that lost flow from those that gained it. On the struck channel this makes it worse than persistence (skill -0.08). On the siblings, which cool by a median of 1.5 K, the same forecast has the right sign in 95 of 98 cases but captures only 31% of the change. A summary score can therefore look acceptable while the model has not registered the event at all.

The gap is caused by the event and by missing information. Two checks rule out the obvious alternatives. First, the gap appears only when the event does (Figure 2). We start forecasts 10, 30, 60 and 110 s before the closure and compare the per-second sibling error of FROZEN and TRUE-FLOW within the same rollout. Averaged over the seconds before the event, FROZEN is worse by 0.005 to 0.04 K; averaged over the seconds after it, by 0.68 to 0.92 K (for the windows that extend at least 60 s past the event). On setpoint events, which do not redistribute flow between channels, FROZEN is worse than TRUE-FLOW over all nodes by 0.009 K at 60 s ($n = 10$), against 0.25 K for closures (per-event means in both cases). Second, the gap is not a property of how the FROZEN model was trained. Giving the TRUE-FLOW model frozen flow at test time raises its sibling error from 0.032

Table 2: **Under FROZEN every model forecasts the wrong direction on the struck channel; with the command, AdvGNODE gets its own channel right but not the siblings.** For each test closure we compare the predicted change $\hat{T}(t_k + 60) - T(t_k)$ with the true one. "Right sign" counts cases with the true sign, per seed and averaged over seeds; "ratio" is the median over events of predicted over true change (1 is ideal). True median changes: $+4.3$ K on the struck channel, -1.5 K on the siblings. Seeds as in Table 1.

		Struck channel (49)		Sibling channels (98)	
Model	Regime	Right sign	Ratio	Right sign	Ratio
AdvGNODE	TRUE-FLOW	49	1.00	98	1.01
AdvGNODE	FROZEN	0	-0.11	95	0.31
AdvGNODE	ESTIMATED	49	1.00	98	1.01
AdvGNODE	COMMAND	46	0.86	96	0.35
Base	TRUE-FLOW	49	1.03	98	0.90
Base	FROZEN	2	-0.10	93	0.30
STGCN	TRUE-FLOW	49	1.01	98	0.95
STGCN	FROZEN	1	-0.09	93	0.27

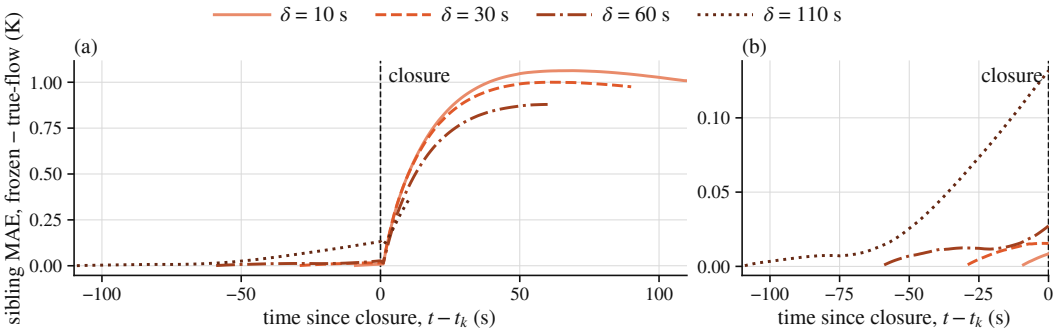


Figure 2: **The gap between FROZEN and TRUE-FLOW appears at the event, not before it.** Per-second mean absolute error on the sibling channels, FROZEN minus TRUE-FLOW (AdvGNODE, mean over three seeds), for forecasts that start 10, 30, 60 and 110 s before the closure; the horizontal axis is time since the event. (a) Whole window; (b) the seconds before the event, enlarged. Before the event the gap averages 0.005–0.04 K and grows slowly with the lead time, because frozen flow misses the slow drift of the controlled flows. At the event it jumps: averaged over the seconds after it, the gap is 0.68–0.92 K for the windows that extend at least 60 s past the event. STGCN behaves the same way (Appendix C).

to 0.99 K and its struck-channel error from 0.085 to 4.04 K, essentially the FROZEN model's errors. Removing the information alone reproduces the failure.

6.2 RECOVERING THE MISSING FLOW

The closed-form estimate recovers almost all of it. Under ESTIMATED, the sibling error is 0.042 K against 1.013 K under FROZEN and 0.032 K under TRUE-FLOW: 99% of the gap. The remaining difference from TRUE-FLOW is statistically clear ($+0.010$ K, $p = 7.6 \times 10^{-12}$) but small, consistent with the estimator's 1.3% median error in the sibling flow change. Directions and magnitudes match TRUE-FLOW (Table 2).

Giving the network the valve command instead does not. With the command, AdvGNODE recovers 6% of the sibling gap (6.5% and 5.8% for the two seeds): its sibling error is 0.972 K. On the struck channel, however, it recovers 85%, from 4.02 to 0.86 K, and it gets the direction right in 46 of 49 events. Table 2 shows the split sharply: with the command, the model's predicted struck-channel change reaches 86% of the true change, while its sibling change reaches 35%, barely more than the 31% of FROZEN. ESTIMATED is better than COMMAND on the siblings in each seed at 30 and 60 s

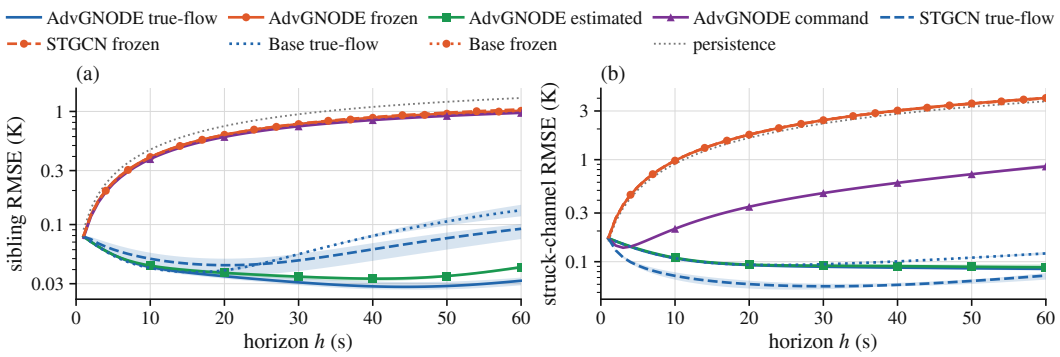


Figure 3: **With true flow, AdvGNODE's sibling error barely changes between 30 and 60 s while STGCN's nearly doubles; without it, all models have nearly the same large error.** RMSE over the first h seconds after the closure against h: (a) sibling channels, (b) struck channel; 49 test closures, logarithmic scale. Colour marks the regime and line style the model (AdvGNODE solid, STGCN dashed, base dotted), with persistence in grey; line: seed mean, band: seed range. At $h = 30$ and 60 s, AdvGNODE under TRUE-FLOW is at 0.031 and 0.032 K on the siblings, while STGCN rises from 0.048 to 0.092 K.

(at 60 s: $p < 10^{-16}$, t-test; $p < 10^{-14}$, Wilcoxon), and also on the struck channel at both horizons ($p \leq 2.4 \times 10^{-4}$).

So from 392 training closures the network learns what a valve does to its own channel, but not what it does to the channels beside it. The redistribution onto the siblings is precisely the part the hydraulic estimate computes; Section 7 explains why flow-following message passing cannot carry it from a local command. The interpretation of the COMMAND regime was fixed before its results were available, and this outcome is the one we had specified as evidence that the estimator adds information the network does not extract from the command.

6.3 WHERE FLOW STRUCTURE HELPS

With identical information, building flow into the messages matters. Base, which receives the same flows as an ordinary input, has a sibling error of 0.134 K under TRUE-FLOW against 0.032 K for AdvGNODE (paired difference -0.086 K over two seeds, $p = 2.2 \times 10^{-16}$), and all-node error 0.103 against 0.067 K. The comparison with STGCN is more mixed. AdvGNODE is better on the siblings (0.032 against 0.092 K) and on the rest of the struck cabinet (0.033 against 0.113 K), while STGCN is better on the four other blocks (0.039 against 0.062 K) and in rollouts without events (Appendix C). STGCN's struck-channel error is lower (0.073 against 0.085 K), but the paired difference is not significant ($p = 0.41$). Over all nodes the paired per-event difference is 2% ($p = 0.58$); in pooled RMSE it is 5% (0.067 against 0.071 K). Between 30 and 60 s, STGCN's sibling error nearly doubles while AdvGNODE's barely changes (Figure 3). STGCN also reads the two samples before t_0, which gives it the current rate of change for free, and this plausibly explains its advantage on slow drift away from the event. Where the dynamics are driven by a change in flow, the flow-structured model is clearly better.

7 DISCUSSION AND LIMITATIONS

The split between the struck channel (85% recovered) and its siblings (6%) follows from how information moves through AdvGNODE. The command enters at the struck channel's node, and messages travel along the flow, one edge per evaluation of the derivative. A sibling shares no edge with the struck channel; the two meet only at the splitter upstream and the merge downstream. By construction, a sibling's temperature derivative does not depend on the struck channel's command except through temperatures carried around the CDU loop. Hydraulic redistribution, in contrast, is a pressure effect felt across the cabinet within seconds, and it is exactly the non-local part the estimator computes. Our result thus shows that flow-following (upwind) message passing does not learn the redistribution from a local command, not that no network could; giving the command to

the whole cabinet, messages against the flow, or a flow solve inside the model are alternatives we did not test.

The cabinet consists of linear resistances and valves (branch fits have $R^2 > 0.9999$, and the identified valve coefficient equals the nominal one to machine precision), so a two-terminal linear model is almost exact. Most of its error comes from representing the rest of the plant by one effective resistance per block (R^2 between 0.962 and 0.973). Real valves are nonlinear and hysteretic, and virtual flow meters in buildings report errors of up to about 10% (Kim et al., 2025). Our 99% recovery is what an estimate with a 1.3% median error in the sibling flow change buys; how forecast error grows as the estimate degrades is open. The failure under FROZEN does not rely on linearity: the struck channel's flow falls whenever its valve closes, and a surrogate given the pre-event flow cannot see it. How strongly the siblings respond depends on how firmly the rest of the plant holds the cabinet's pressure difference; here the CDU pump controller acts over minutes, so the siblings absorb most of the lost flow. FROZEN and ESTIMATED also assume that every branch flow is known when the forecast starts. Here one flow measurement per cabinet and the valve positions suffice, because the branch model is exact; with nonlinear valves this needs a calibrated branch model or a meter on each branch.

All results come from one simulated plant, whose absolute secondary-side pressures are unphysical; we use only pressure differences. We test single-channel closures to 10–50% of a base opening between 0.3 and 1.0, without sensor noise. Valve-opening events, simultaneous closures, actuation outside the training range and nonlinear valves remain untested. The protocol needs only a simulator with valve inputs and logged branch flows, so it can be repeated on other plants, including ones with nonlinear valves. Our 30–60 s horizons are short compared with the roughly 500 s the temperatures take to settle, but the gap between FROZEN and TRUE-FLOW does not close within our windows: averaged over the 110 s after the event it is 0.92 K, against 0.68 K over the first 60 s (Figure 2), so the cost does not fade after the first minute. Finally, inputs are held at their start-of-step value, so in every regime, TRUE-FLOW included, the post-event flow enters one 1 s step late.

Moreover, AdvGNODE and STGCN have three seeds per regime, COMMAND two, and base two in both. Paired tests are uncorrected; every sibling-channel comparison between regimes has $p < 10^{-9}$, while STGCN against AdvGNODE on the struck channel is not significant. The event-locked, same-weights and setpoint analyses were designed after the primary results and are diagnostics based on absolute errors. Before the event FROZEN is already worse than TRUE-FLOW by 0.005–0.04 K, because frozen flow misses the slow drift of the controlled flows (Appendix C); the evidence that the event causes the gap is the jump to 0.68–0.92 K when it occurs.

8 CONCLUSION

A learned temperature surrogate that is not told how flow redistributed after a valve moved does not merely lose accuracy: it forecasts the wrong direction on the closed channel and misses most of the change on its siblings, whatever its architecture. On a simulator of the Frontier cooling plant, withholding post-event flow raised AdvGNODE's sibling-channel error about 32-fold. A standard closed-form flow estimate from pre-event measurements and the valve command recovered 99% of that loss. The command given directly to the network recovered 85% on the closed channel but only 6% on its siblings, which flow-following message passing does not connect to the struck channel's command. Surrogates that forecast through actuation should be given the post-event flow field, and where a reliable branch model exists it can be computed rather than measured.

AI USE STATEMENT

We used Grammarly AI tool only to edit the manuscript for readability, which included rephrasing sentences, reducing repetition, and converting the text to American English spelling and punctuation. We did not use generative AI tools to generate data, formulate hypotheses, design experiments or methodology, implement methods or write code, analyze or interpret results, or produce figures. All scientific content, experiments, numbers, and conclusions are our own. The authors reviewed every AI-assisted edit against the original text to confirm that no numbers, claims, or citations were altered.

Reproducibility statement

The plant model is the publicly available simulator distributed with LC-Opt (Naug et al., 2025). Sections 3–5 and the appendix give what is needed to interpret and check the results: the operating conditions, events and splits and the verification of the graph (Appendix A), the redistribution estimator and its accuracy (Section 4.1 and Appendix A), and the information each model receives, the shared training schedule and the checkpoint-selection rule (Section 5 and Appendix B). Each run is deterministic given its seed, and every number in the paper is produced by a script from the stored evaluation outputs. We will release the code, the generated data and the trained models after acceptance.

References

Akzhol Almukhametov, Doyeong Lim, Rui Hu, and Yang Liu. Graph neural ODE digital twins for control-oriented reactor thermal-hydraulic forecasting under partial observability. *arXiv preprint arXiv:2604.07292*, 2026.

Timur Bikmukhametov and Johannes Jäschke. First principles and machine learning virtual flow metering: A literature review. *Journal of Petroleum Science and Engineering*, 184:106487, 2020. doi: 10.1016/j.petrol.2019.106487.

Laura Boca de Giuli, Alessio La Bella, and Riccardo Scattolini. Physics-informed neural network modeling and predictive control of district heating systems. *IEEE Transactions on Control Systems Technology*, 32(4):1182–1195, 2024. doi: 10.1109/TCST.2024.3355476.

Roberto Boghetti, Jean-Marc Odobez, and Jérôme Henri Kämpf. Graph neural network-based surrogate modeling for fast and scalable simulations of meshed district heating networks. *Energy and AI*, 24:100721, 2026. doi: 10.1016/j.egyai.2026.100721.

Andreas Bott, Mario Beykirch, and Florian Steinke. Efficient training of learning-based thermal power flow for 4th generation district heating grids. *Energy*, 318:134661, 2025. doi: 10.1016/j.energy.2025.134661.

Taha Boussaid, François Rousset, Vasile-Marian Scuturici, and Marc Clausse. Enabling fast prediction of district heating networks transients via a physics-guided graph neural network. *Applied Energy*, 370:123634, 2024. doi: 10.1016/j.apenergy.2024.123634.

Johannes Brandstetter, Daniel E. Worrall, and Max Welling. Message passing neural PDE solvers. In *International Conference on Learning Representations*, 2022.

Wesley Brewer, Matthias Maiterth, Vineet Kumar, Rafal Wojda, Sedrick Bouknight, Jesse Hines, Woong Shin, Scott Greenwood, David Grant, Wesley Williams, and Feiyi Wang. A digital twin framework for liquid-cooled supercomputers as demonstrated at exascale. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis (SC)*, 2024. doi: 10.1109/SC41406.2024.00029.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K. Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, 2018.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *International Conference on Machine Learning*, pp. 1263–1272, 2017.

Scott Greenwood, Vineet Kumar, and Wesley Brewer. Thermo-fluid modeling framework for supercomputer digital twins: Part 2, automated cooling models. In *Proceedings of the American Modelica Conference*, pp. 208–217, 2024. doi: 10.3384/ecp207208.

Hao Hu and Xue-shan Ai. Exogenous dropout: A simple, strong baseline for corruption-robust time series forecasting with covariates. *arXiv preprint arXiv:2607.05452*, 2026.

Shrenik Jadhav and Zheng Liu. Machine learning guided cooling system optimization for data center. *arXiv preprint arXiv:2601.02275*, 2026.

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

Taeyang Kim, Hyojun Kim, Jinhyun Lee, and Younghum Cho. Development of virtual water flow sensor using valve performance curve. *Journal of Sensor and Actuator Networks*, 14(1):1, 2025. doi: 10.3390/jsan14010001.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Vineet Kumar, Scott Greenwood, Wesley Brewer, David Grant, Nathan Parkison, and Wesley Williams. Thermo-fluid modeling framework for supercomputer digital twins: Part 1, demonstration at exascale. In *Proceedings of the American Modelica Conference*, pp. 199–207, 2024. doi: 10.3384/ecp207199.

Allan H. Murphy. Skill scores based on the mean square error and their relationships to the correlation coefficient. *Monthly Weather Review*, 116(12):2417–2424, 1988.

Avisek Naug, Antonio Guillen-Perez, Vineet Kumar, Scott Greenwood, Wesley Brewer, Sahand Ghorbanpour, Ashwin Ramesh Babu, Vineet Gundecha, Ricardo Luna Gutierrez, and Soumyendu Sarkar. LC-Opt: Benchmarking reinforcement learning and agentic AI for end-to-end liquid cooling optimization in data centers. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, volume 38, 2025. arXiv:2511.00116.

Yishak Tadele Nigatu, Wesley Brewer, Jonathan McConnell, Valentine G. Anantharaj, and Sandro Fiore. Foundations for FMU surrogate model design in ExaDigiT: Preliminary analysis of three IBM POWER9 cooling systems. In *Proceedings of the 2nd International Workshop on Systems and Methods for Sustainable Large-Scale AI (GreenSys '26)*, pp. 31–36. ACM, 2026. doi: 10.1145/3802973.3804456.

Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W. Battaglia. Learning mesh-based simulation with graph networks. In *International Conference on Learning Representations*, 2021.

Michael Poli, Stefano Massaroli, Junyoung Park, Atsushi Yamashita, Hajime Asama, and Jinkyoo Park. Graph neural ordinary differential equations. *arXiv preprint arXiv:1911.07532*, 2019.

Erfan Shakhesi, Maurice Heemels, and Paula Chanfreut. Model predictive control for dynamic hydraulic balancing in building radiator heating systems: Modeling, design, and experimental validation. *arXiv preprint arXiv:2609.08708*, 2026.

Li Song, Gang Wang, and Michael R. Brambley. Uncertainty analysis for a virtual flow meter using an air-handling unit chilled water valve. *HVAC&R Research*, 19(3):335–345, 2013. doi: 10.1080/10789669.2013.774890.

Yogesh Verma, Markus Heinonen, and Vikas Garg. ClimODE: Climate and weather forecasting with physics-informed neural ODEs. In *International Conference on Learning Representations*, 2024.

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

Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 3634–3640, 2018.

A PLANT, DATA AND GRAPH

Plant model. The plant is a Functional Mock-up Unit (FMI 2.0) of the Frontier liquid-cooling system from the ExaDigiT project (Brewer et al., 2024; Kumar et al., 2024), as distributed with LC-Opt (Naug et al., 2025). Five compute blocks share a central plant. Each block has a CDU and a cabinet whose splitter feeds three parallel cooling channels, each a flow resistance, a channel volume heated by the IT load and a linear valve; channel 3 has one additional resistance. Resistances and valves are linear in the pressure difference, which is why the branch model of Section 4.1 is almost exact (Section 7). The CDU pump speed follows a PI controller on the cabinet-loop pressure difference through a slow first-order filter (time constant about 159 s), so in the first tens of seconds after a closure the cabinet's pressure difference is not held and the siblings absorb most of the lost flow. Absolute pressures on the secondary side are negative in most samples, a defect of the model; we use only pressure differences.

Operating conditions, events and splits. Base valve openings are drawn from $\mathcal{U}[0.3, 1.0]$ and the setpoints of each CDU from the environment's operating ranges, $\mathcal{U}[25, 38]$ for the pressure difference (in psi, the environment's unit) and $\mathcal{U}[20, 40]$ °C for the supply temperature; IT load and outdoor wet-bulb temperature follow recorded profiles. Of the 700 rollouts, 490 contain a closure, 105 a step of one CDU's pressure-difference setpoint and 105 no event. They are split by rollout, stratified by type, into training, validation and test (closure 392/49/49, setpoint 84/11/10, none 84/10/11). Events fall between 320 and 380 s, so that every evaluation window, including those that start 110 s before the event, lies after the 200 s relaxation period.

Graph verification. At every volume that exposes its own accumulation term, the signed sum of incident edge flows matches that term to a relative error below 10^{-12} at every logged step, in a closure rollout and in a rollout without events. The seven boundary volumes excluded from the surrogates, which impose pressure or absorb thermal expansion, together exchange 0.135% of the IT heat input and stay within 0.153 K over $t \geq 200$ s.

Sibling-absorbed fraction. Over the 392 training closures, the siblings absorb 69–92% (mean 81%) of the flow the struck channel loses. The identified estimator implies 69–87% (mean 80%) on the same events, with a median absolute error of 0.010. At base openings 0.555/0.226/0.007 the measured fraction is 0.44, so it is a property of the operating point, not of the plant.

Redistribution estimator. Channel c of a cabinet has conductance $G_c = \big[R_c + 1/(o_c k) + R_6 \mathbb{1}[c = 3]\big]^{-1}$, so that $\dot{m}_c = G_c \, \Delta p$ across the cabinet. The branch resistances and the valve coefficient are identified from pre-event steady states of the training rollouts; every branch fit has $R^2 > 0.9999$, and the valve coefficient equals its nominal value to machine precision. Seen from the splitter and merge, the rest of the plant is a Thévenin equivalent, $\Delta p = p_s - R_{\text{ext}} \sum_c \dot{m}_c$. One effective R_{ext} per block is identified from that block's training closures by comparing each event's pre-event state with the window $[t_k + 20, t_k + 30]$ s (fit R^2 between 0.962 and 0.973), so it includes the pump controller's response at that horizon. At forecast time, $p_s = \Delta p_{\text{pre}} + R_{\text{ext}} \sum_c \dot{m}_{c,\text{pre}}$ is computed from the event's own pre-event state, the post-event openings give new conductances G'_c, and $\Delta p' = p_s / (1 + R_{\text{ext}} \sum_c G'_c)$, $\dot{m}'_c = G'_c \, \Delta p'$. The change is carried into the same block's CDU loop by mass balance, with no further fitted parameters. The estimator uses no test data and no post-event measurement. Setpoint events receive no estimate, so ESTIMATED equals FROZEN on them. On the 49 validation closures, the median relative error of the estimated flow change is 0.15% on the struck channel (90th percentile 0.40%), 1.29% on the siblings (2.49%) and 6.59% on the cabinet total (14.2%).

B MODELS AND TRAINING

Information available to each model. Within a regime, all learned models receive the same information: node temperatures at t_0, heat inputs, node types and the regime's flow. AdvGNODE and base differ only in how flow enters the stream messages: AdvGNODE multiplies each message by the edge's normalised flow, so an edge without flow carries no message, while base adds the flow to the message input (72,559 and 72,577 parameters). STGCN, adapted from Yu et al. (2018) (37,569

parameters), receives each node's summed incoming flow and is the only model that also reads the two samples before t_0. In the COMMAND regime each cooling-channel node of AdvGNODE also receives its own commanded opening and every other node receives 0; messages still travel only along the flow (Section 7). This adds 64 parameters. A variant of AdvGNODE without any flow input served as a diagnostic and is not reported.

Training and selection. All learned models share the training schedule of Section 5 and one selection rule: validation skill at 60 s on a fixed set of 300 validation windows, smoothed by a centred three-epoch moving average. Test data are never used for training or selection. Each run is deterministic given its seed. Table 3 lists the runs and Figure 6 shows their validation curves.

Table 3: **Every completed run selected a checkpoint between epochs 92 and 99, and no run produced a NaN.** Runs evaluated in this paper, with the selected epoch and the smoothed validation skill at 60 s for each seed (seeds 0, 1, 2 in order). The selection set contains windows that start before the event, so FROZEN runs score lower on it.

Model	Regime	Seeds	Parameters	Selected epoch	Validation skill
AdvGNODE	TRUE-FLOW	3	72,559	97, 99, 97	0.885, 0.881, 0.883
AdvGNODE	FROZEN	3	72,559	96, 99, 98	0.693, 0.689, 0.691
AdvGNODE	ESTIMATED	3	72,559	95, 99, 96	0.877, 0.877, 0.880
AdvGNODE	COMMAND	2	72,623	99, 99	0.875, 0.841
Base	TRUE-FLOW	2	72,577	96, 95	0.820, 0.854
Base	FROZEN	2	72,577	99, 92	0.683, 0.664
STGCN	TRUE-FLOW	3	37,569	98, 98, 99	0.920, 0.924, 0.926
STGCN	FROZEN	3	37,569	99, 99, 98	0.707, 0.708, 0.709

C ADDITIONAL RESULTS

Full results. Tables 4 and 5 give pooled RMSE for all five node sets at 60 and 30 s, with the range over seeds. Table 6 gives the paired tests. Their means are means over events of per-event RMSE, which differ from the pooled RMSE of Table 1.

Recovery. For a regime X, the recovery is $R_X = \sum_e (\epsilon_e^{\text{FROZEN}} - \epsilon_e^X) / \sum_e (\epsilon_e^{\text{FROZEN}} - \epsilon_e^{\text{TRUE-FLOW}})$, where ϵ_e is the RMSE of test closure e averaged over the seeds that X, FROZEN and TRUE-FLOW share (Table 7). At 60 s, ESTIMATED recovers 98.6–99.2% of the sibling gap in every seed; COMMAND recovers 5.8% and 6.5% on the siblings and 83–86% on the struck channel. On the siblings, FROZEN is worse than TRUE-FLOW by a factor of about 32 for AdvGNODE (pooled RMSE, 1.0127/0.0319; 31.1 in per-event means) and 11.4 for STGCN (pooled RMSE).

Event-locked control. Table 8 lists the per-second mean absolute error differences before and after the event for forecasts that start δ seconds before it. For both architectures the difference between FROZEN and TRUE-FLOW is 0.001–0.04 K before the event and 0.2–4.5 K after it. The difference between ESTIMATED and TRUE-FLOW is at most 0.11 K after the event.

Same weights, different flow. Table 9 evaluates each checkpoint with the other regime's flow. Given frozen flow, the TRUE-FLOW model's sibling error rises from 0.032 to 0.991 K, close to the FROZEN model's 1.013 K. Given true flow, the FROZEN model recovers most of its struck-channel error (0.110 K) but only part of its sibling error (0.394 K). In control windows, which end before the event, frozen flow still costs the TRUE-FLOW model 0.019 K on the siblings ($p = 1.5 \times 10^{-10}$), the effect of the slow drift of the controlled flows, while the FROZEN model is insensitive to the flow it is given ($p = 0.76$).

Setpoint events. A setpoint step changes a cabinet's total flow but not its split between channels. On the 10 test setpoint events, FROZEN is worse than TRUE-FLOW over all nodes at 60 s by 0.0087 K for AdvGNODE ($p = 6.5 \times 10^{-4}$, t-test; $p = 3.9 \times 10^{-3}$, Wilcoxon) and 0.0063 K for STGCN, against 0.25 K and 0.24 K on the 49 test closures (per-event means).

Rollouts without events. Table 10 reports the 11 test rollouts without events. STGCN, which reads the recent history, is the most accurate here.

Table 4: **The cost of missing flow is concentrated in the struck cabinet; the other four blocks change little.** Pooled RMSE (K) over the first 60 s after the closure, 49 test closures; seed mean, with the range over seeds below it. Bold: lowest in each column.

Model	Regime	Seeds	Struck	Siblings	Rest of cabinet	Other blocks	All nodes
Persistence	–	–	3.7299	1.3149	0.6280	0.3051	0.4343
AdvGNODE	TRUE-FLOW	3	0.0851 0.0819–0.0898	**0.0319** 0.0290–0.0340	**0.0331** 0.0285–0.0355	0.0623 0.0605–0.0641	**0.0669** 0.0650–0.0691
AdvGNODE	FROZEN	3	4.0209 4.0201–4.0213	1.0127 1.0124–1.0130	0.3994 0.3988–0.4004	0.0721 0.0699–0.0753	0.3375 0.3371–0.3380
AdvGNODE	ESTIMATED	3	0.0882 0.0837–0.0944	0.0421 0.0404–0.0432	0.0488 0.0453–0.0527	0.0690 0.0654–0.0726	0.0705 0.0677–0.0726
AdvGNODE	COMMAND	2	0.8589 0.8181–0.8997	0.9719 0.9699–0.9740	0.3446 0.3423–0.3468	0.0643 0.0559–0.0726	0.1580 0.1526–0.1634
Base	TRUE-FLOW	2	0.1205 0.1166–0.1245	0.1339 0.1183–0.1496	0.1553 0.1380–0.1727	0.0926 0.0830–0.1022	0.1034 0.0919–0.1149
Base	FROZEN	2	4.0146 4.0108–4.0185	1.0132 1.0114–1.0151	0.4101 0.4093–0.4109	0.1031 0.0947–0.1115	0.3433 0.3413–0.3453
STGCN	TRUE-FLOW	3	**0.0731** 0.0667–0.0775	0.0917 0.0745–0.1009	0.1130 0.0991–0.1305	**0.0392** 0.0372–0.0413	0.0706 0.0684–0.0750
STGCN	FROZEN	3	3.9922 3.9878–3.9980	1.0425 1.0389–1.0445	0.4058 0.3972–0.4114	0.0468 0.0427–0.0510	0.3344 0.3333–0.3355

Table 5: **The same pattern holds at 30 s.** As Table 4, over the first 30 s.

Model	Regime	Seeds	Struck	Siblings	Rest of cabinet	Other blocks	All nodes
Persistence	–	–	2.2631	0.9444	0.4006	0.1801	0.2672
AdvGNODE	TRUE-FLOW	3	0.0892 0.0882–0.0912	**0.0308** 0.0292–0.0336	**0.0233** 0.0204–0.0265	0.0334 0.0330–0.0339	**0.0429** 0.0424–0.0436
AdvGNODE	FROZEN	3	2.4357 2.4348–2.4363	0.7731 0.7726–0.7735	0.2892 0.2888–0.2898	0.0352 0.0347–0.0356	0.2141 0.2140–0.2142
AdvGNODE	ESTIMATED	3	0.0914 0.0872–0.0943	0.0346 0.0330–0.0356	0.0299 0.0274–0.0315	0.0343 0.0336–0.0355	0.0438 0.0432–0.0445
AdvGNODE	COMMAND	2	0.4703 0.4699–0.4707	0.7388 0.7369–0.7407	0.2067 0.2039–0.2095	0.0329 0.0304–0.0355	0.1059 0.1047–0.1072
Base	TRUE-FLOW	2	0.0945 0.0934–0.0955	0.0547 0.0539–0.0556	0.0855 0.0837–0.0873	0.0498 0.0474–0.0523	0.0661 0.0618–0.0704
Base	FROZEN	2	2.4361 2.4349–2.4373	0.7704 0.7697–0.7711	0.2910 0.2909–0.2912	0.0538 0.0503–0.0573	0.2166 0.2161–0.2172
STGCN	TRUE-FLOW	3	**0.0574** 0.0538–0.0609	0.0482 0.0388–0.0551	0.0771 0.0644–0.0970	**0.0171** 0.0166–0.0175	0.0520 0.0504–0.0541
STGCN	FROZEN	3	2.4297 2.4275–2.4319	0.7776 0.7764–0.7790	0.2896 0.2867–0.2923	0.0216 0.0191–0.0239	0.2126 0.2121–0.2130

Table 6: **Every sibling-channel difference is significant at $p < 10^{-9}$; AdvGNODE and STGCN under TRUE-FLOW do not differ on the struck channel or over all nodes.** Paired tests over the 49 test closures. Each event's RMSE is averaged over the seeds shown, then paired; $\bar{a}$ and $\bar{b}$ are means over events of per-event RMSE (K), and $\bar{d}$ is the mean of $a - b$ (negative: a is more accurate). t: paired t-test; W: Wilcoxon signed-rank; both two-sided and uncorrected. With 49 events the smallest possible Wilcoxon p is 3.55×10^{-15}, reached only when every event differs in the same direction.

Seeds	Node set	h (s)	$\bar{a}$	$\bar{b}$	$\bar{d}$	p (t)	p (W)
AdvGNODE TRUE-FLOW *vs. AdvGNODE* FROZEN							
3	sibling	60	0.0296	0.9204	-0.8908	1.61×10^{-19}	3.55×10^{-15}
3	struck	60	0.0700	3.5630	-3.4930	1.16×10^{-17}	3.55×10^{-15}
3	all nodes	60	0.0637	0.3106	-0.2469	1.82×10^{-18}	3.55×10^{-15}
AdvGNODE ESTIMATED *vs. AdvGNODE* FROZEN							
3	sibling	60	0.0397	0.9204	-0.8806	3.45×10^{-19}	3.55×10^{-15}
3	struck	60	0.0740	3.5630	-3.4891	1.20×10^{-17}	3.55×10^{-15}
3	all nodes	60	0.0678	0.3106	-0.2428	3.62×10^{-18}	3.55×10^{-15}
AdvGNODE ESTIMATED *vs. AdvGNODE* TRUE-FLOW							
3	sibling	60	0.0397	0.0296	$+0.0101$	7.57×10^{-12}	1.53×10^{-11}
3	struck	60	0.0740	0.0700	$+0.0039$	0.0245	9.13×10^{-3}
3	all nodes	60	0.0678	0.0637	$+0.0041$	3.20×10^{-8}	6.15×10^{-9}
AdvGNODE COMMAND *vs. AdvGNODE* FROZEN							
2	sibling	60	0.8656	0.9202	-0.0545	1.36×10^{-10}	8.44×10^{-12}
2	struck	60	0.6088	3.5634	-2.9546	2.04×10^{-15}	3.55×10^{-15}
2	all nodes	60	0.1494	0.3105	-0.1612	1.17×10^{-11}	3.55×10^{-15}
AdvGNODE COMMAND *vs. AdvGNODE* TRUE-FLOW							
2	sibling	60	0.8656	0.0305	$+0.8351$	1.16×10^{-17}	3.55×10^{-15}
2	struck	60	0.6088	0.0685	$+0.5403$	4.21×10^{-8}	3.55×10^{-15}
2	all nodes	60	0.1494	0.0637	$+0.0857$	1.79×10^{-14}	3.55×10^{-15}
AdvGNODE COMMAND *vs. AdvGNODE* ESTIMATED							
2	sibling	30	0.6614	0.0325	$+0.6289$	4.48×10^{-18}	3.55×10^{-15}
2	struck	30	0.2920	0.0812	$+0.2108$	2.41×10^{-4}	6.97×10^{-10}
2	all nodes	30	0.0999	0.0414	$+0.0586$	8.32×10^{-15}	3.55×10^{-15}
2	sibling	60	0.8656	0.0395	$+0.8261$	2.50×10^{-17}	3.55×10^{-15}
2	struck	60	0.6088	0.0757	$+0.5331$	5.28×10^{-8}	3.55×10^{-15}
2	all nodes	60	0.1494	0.0692	$+0.0802$	1.37×10^{-13}	3.55×10^{-15}
STGCN TRUE-FLOW *vs. STGCN* FROZEN							
3	sibling	60	0.0804	0.9503	-0.8700	1.34×10^{-18}	3.55×10^{-15}
3	struck	60	0.0639	3.5323	-3.4684	2.40×10^{-17}	3.55×10^{-15}
3	all nodes	60	0.0650	0.3066	-0.2416	2.71×10^{-19}	3.55×10^{-15}
Base TRUE-FLOW *vs. Base* FROZEN							
2	sibling	60	0.1170	0.9224	-0.8055	1.52×10^{-18}	3.55×10^{-15}
2	struck	60	0.1030	3.5573	-3.4543	1.74×10^{-17}	3.55×10^{-15}
2	all nodes	60	0.0991	0.3174	-0.2183	6.79×10^{-18}	3.55×10^{-15}
AdvGNODE TRUE-FLOW *vs. Base* TRUE-FLOW							
2	sibling	60	0.0305	0.1170	-0.0864	2.18×10^{-16}	3.55×10^{-15}
2	struck	60	0.0685	0.1030	-0.0346	2.80×10^{-7}	1.53×10^{-11}
2	all nodes	60	0.0637	0.0991	-0.0354	2.13×10^{-21}	3.55×10^{-15}
AdvGNODE TRUE-FLOW *vs. STGCN* TRUE-FLOW							
3	sibling	60	0.0296	0.0804	-0.0508	2.06×10^{-12}	3.55×10^{-15}
3	struck	60	0.0700	0.0639	$+0.0061$	0.408	0.851
3	all nodes	60	0.0637	0.0650	-0.0014	0.578	0.541
AdvGNODE FROZEN *vs. STGCN* FROZEN							
3	sibling	60	0.9204	0.9503	-0.0300	3.86×10^{-10}	4.04×10^{-11}
3	struck	60	3.5630	3.5323	$+0.0307$	1.66×10^{-8}	4.65×10^{-9}
3	all nodes	60	0.3106	0.3066	$+0.0039$	5.02×10^{-9}	2.38×10^{-9}

Table 7: **ESTIMATED closes about 99% of the gap on both node sets; COMMAND closes most of it on the struck channel but about 6% on the siblings.** Recovery R_X per seed and over matched seeds. The two middle columns are means over the 49 test closures of per-event RMSE differences (K).

X	Node set	h (s)	Seeds	$\overline{\epsilon^{\text{FROZEN}} - \epsilon^{X}}$	$\overline{\epsilon^{\text{FROZEN}} - \epsilon^{\text{TRUE-FLOW}}}$	R_X
ESTIMATED	sibling	30	seed 0	0.6676	0.6720	99.4%
ESTIMATED	sibling	30	seed 1	0.6646	0.6674	99.6%
ESTIMATED	sibling	30	seed 2	0.6665	0.6715	99.3%
ESTIMATED	sibling	30	3-seed mean	0.6662	0.6703	**99.4%**
ESTIMATED	sibling	60	seed 0	0.8796	0.8903	98.8%
ESTIMATED	sibling	60	seed 1	0.8816	0.8890	99.2%
ESTIMATED	sibling	60	seed 2	0.8806	0.8930	98.6%
ESTIMATED	sibling	60	3-seed mean	0.8806	0.8908	**98.9%**
ESTIMATED	struck	30	seed 0	2.1131	2.1177	99.8%
ESTIMATED	struck	30	seed 1	2.1165	2.1214	99.8%
ESTIMATED	struck	30	seed 2	2.1199	2.1168	100.2%
ESTIMATED	struck	30	3-seed mean	2.1165	2.1186	**99.9%**
ESTIMATED	struck	60	seed 0	3.4842	3.4926	99.8%
ESTIMATED	struck	60	seed 1	3.4912	3.4973	99.8%
ESTIMATED	struck	60	seed 2	3.4918	3.4891	100.1%
ESTIMATED	struck	60	3-seed mean	3.4891	3.4930	**99.9%**
COMMAND	sibling	30	seed 0	0.0401	0.6720	6.0%
COMMAND	sibling	30	seed 1	0.0344	0.6674	5.1%
COMMAND	sibling	30	2-seed mean	0.0372	0.6697	**5.6%**
COMMAND	sibling	60	seed 0	0.0577	0.8903	6.5%
COMMAND	sibling	60	seed 1	0.0514	0.8890	5.8%
COMMAND	sibling	60	2-seed mean	0.0545	0.8897	**6.1%**
COMMAND	struck	30	seed 0	1.9029	2.1177	89.9%
COMMAND	struck	30	seed 1	1.9050	2.1214	89.8%
COMMAND	struck	30	2-seed mean	1.9040	2.1196	**89.8%**
COMMAND	struck	60	seed 0	2.9939	3.4926	85.7%
COMMAND	struck	60	seed 1	2.9153	3.4973	83.4%
COMMAND	struck	60	2-seed mean	2.9546	3.4949	**84.5%**

Table 8: **For both architectures the gap is a few hundredths of a kelvin before the event and up to 4.5 K after it; ESTIMATED stays within 0.11 K of TRUE-FLOW.** Per-second mean absolute error, difference of seed means (three seeds each), averaged over the n seconds before the event ($t - t_k \leq 0$, excluding the first sample) and after it ($t - t_k \geq 1$), for 120 s windows that start δ seconds before the closure; 49 test closures. Range: over seed pairs.

δ (s)	Node set	Pre: n	Pre: mean (K)	Post: n	Post: mean (K)	Post: range
AdvGNODE: FROZEN−TRUE-FLOW						
10	sibling	10	+0.00501	110	+0.9181	+0.9089–+0.9308
10	struck	10	+0.00368	110	+4.5374	+4.5328–+4.5429
30	sibling	30	+0.01059	90	+0.8511	+0.8436–+0.8590
30	struck	30	+0.00777	90	+4.1051	+4.1004–+4.1098
60	sibling	60	+0.01207	60	+0.6832	+0.6708–+0.6948
60	struck	60	+0.01102	60	+3.2566	+3.2488–+3.2703
110	sibling	110	+0.03924	10	+0.2319	+0.1735–+0.3135
110	struck	110	+0.03741	10	+0.7812	+0.7417–+0.8221
STGCN: FROZEN−TRUE-FLOW						
10	sibling	10	+0.00127	110	+0.9331	+0.9056–+0.9493
10	struck	10	+0.00081	110	+4.4214	+4.4022–+4.4386
30	sibling	30	+0.01143	90	+0.9284	+0.8998–+0.9486
30	struck	30	+0.00813	90	+3.9755	+3.9571–+3.9961
60	sibling	60	+0.02364	60	+0.8138	+0.7666–+0.8443
60	struck	60	+0.02257	60	+3.1013	+3.0719–+3.1448
110	sibling	110	+0.04153	10	+0.2053	+0.1004–+0.3081
110	struck	110	+0.03504	10	+0.6277	+0.5559–+0.7180
AdvGNODE: ESTIMATED−TRUE-FLOW						
10	sibling	10	+0.00007	110	+0.0422	+0.0352–+0.0532
10	struck	10	−0.00032	110	+0.0231	+0.0128–+0.0339
30	sibling	30	+0.00115	90	+0.0489	+0.0401–+0.0614
30	struck	30	+0.00021	90	+0.0258	+0.0150–+0.0361
60	sibling	60	+0.00507	60	+0.0716	+0.0544–+0.0873
60	struck	60	+0.00363	60	+0.0489	+0.0387–+0.0623
110	sibling	110	+0.03141	10	+0.1067	+0.0862–+0.1313
110	struck	110	+0.02899	10	+0.1127	+0.0957–+0.1340

Table 9: **Frozen flow alone reproduces the failure: the TRUE-FLOW model given frozen flow is as wrong as the FROZEN model.** AdvGNODE, pooled RMSE (K) at 60 s, mean over three seeds. Columns give the checkpoint and the flow it receives at test time. Event windows start at the event; control windows start 60 s before it and end at the event.

		TRUE-FLOW checkpoint		FROZEN checkpoint	
Windows	Node set	TRUE-FLOW flow	FROZEN flow	FROZEN flow	TRUE-FLOW flow
event, closure ($n = 49$)	sibling	0.0319	0.9909	1.0127	0.3941
event, closure ($n = 49$)	struck	0.0851	4.0380	4.0209	0.1104
event, closure ($n = 49$)	all nodes	0.0669	0.3385	0.3375	0.0956
event, setpoint ($n = 10$)	all nodes	0.0517	0.0652	0.0594	0.0555
control, closure ($n = 49$)	sibling	0.0155	0.0384	0.0321	0.0329
control, closure ($n = 49$)	struck	0.0138	0.0324	0.0287	0.0274
control, closure ($n = 49$)	all nodes	0.0829	0.0887	0.0886	0.0920
control, setpoint ($n = 10$)	all nodes	0.0995	0.1005	0.1042	0.1096

Table 10: **Without events every model beats persistence, and STGCN, which reads recent history, is the most accurate.** Pooled RMSE (K) over all nodes and skill against persistence on the 11 test rollouts without events; seed means.

Model	Regime	Seeds	RMSE 60 s	RMSE 120 s	Skill 60 s	Skill 120 s
Persistence	–	–	0.2688	0.4342	0.0000	0.0000
AdvGNODE	TRUE-FLOW	3	0.0726	0.1264	0.7299	0.7089
AdvGNODE	FROZEN	3	0.0779	0.1689	0.7101	0.6110
AdvGNODE	ESTIMATED	3	0.0776	0.1613	0.7113	0.6284
STGCN	TRUE-FLOW	3	0.0559	0.1224	0.7919	0.7181
STGCN	FROZEN	3	0.0581	0.1115	0.7840	0.7433
AdvGNODE	COMMAND	2	0.0673	0.1439	0.7496	0.6686
Base	TRUE-FLOW	2	0.1005	0.1891	0.6262	0.5645
Base	FROZEN	2	0.1046	0.2025	0.6110	0.5336

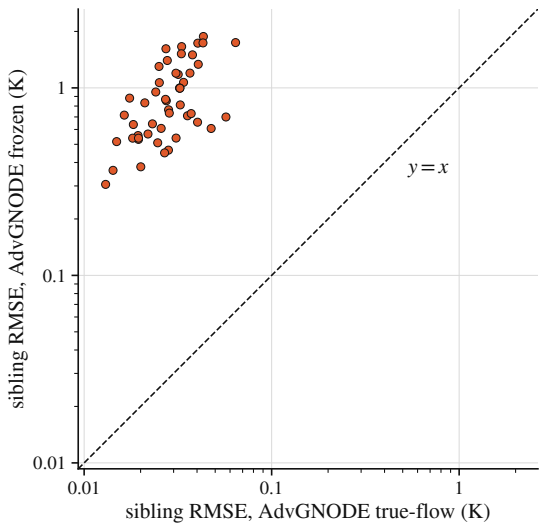


Figure 4: **Every test closure is forecast worse under FROZEN than under TRUE-FLOW.** Per-event sibling-channel RMSE over the first 60 s (AdvGNODE, mean over three seeds), FROZEN against TRUE-FLOW, logarithmic axes; each point is one of the 49 test closures. All points lie above the diagonal, which is what the Wilcoxon $p = 3.55 \times 10^{-15}$ of Table 6 requires.

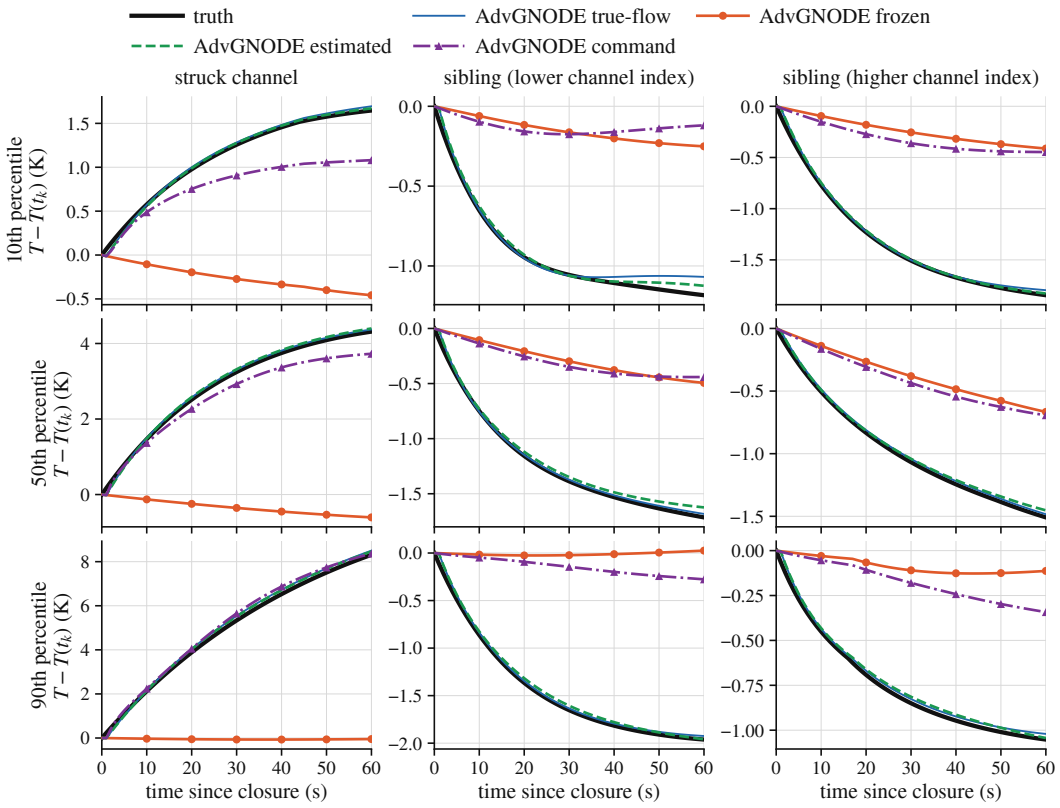


Figure 5: **Under FROZEN the forecast misses the event on the struck channel; under ESTIMATED it follows the truth.** Temperature change since the closure of the struck channel and the two siblings (columns) for the test closures at the 10th, 50th and 90th percentile of the struck channel's true temperature rise over the first 60 s (rows), truth and AdvGNODE forecasts (seed 0) under each regime. Under COMMAND the struck channel moves the right way, but the siblings move far less than the truth. The middle row is the closure shown in Figure 1.

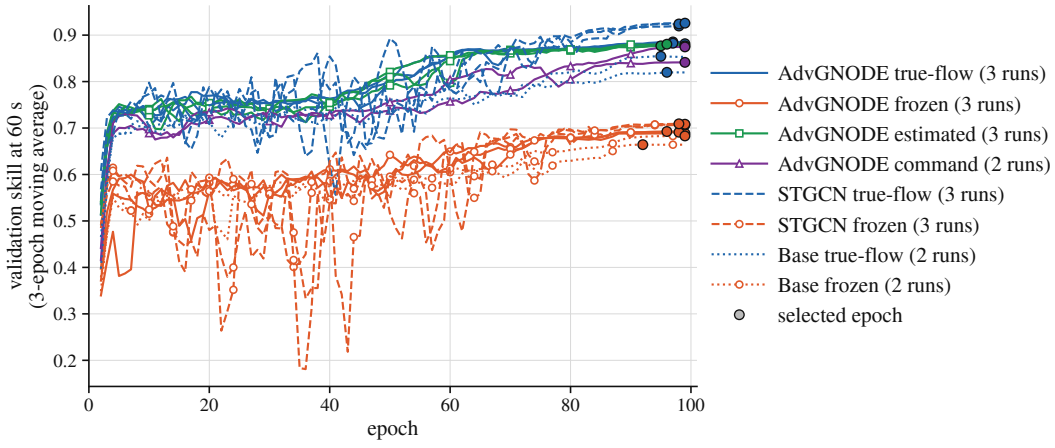


Figure 6: **Every completed run reaches its best smoothed validation skill between epochs 92 and 99.** Validation skill at 60 s (centred three-epoch moving average, so undefined at the first and last epoch) against epoch, one line per run; colour marks the regime and line style the model, and dots mark the selected epochs.