GAPFLOW: CROSS-VARIABLE PRIORS FOR FLOW-MATCHING TIME-SERIES IMPUTATION ON CONTIGUOUS GAPS

Anonymous authors
Paper under double-blind review

ABSTRACT

Flow matching is a strong framework for multivariate time-series imputation. Replacing the Gaussian source with a structured prior that already interpolates the observed entries shortens the transport the network must learn, and yields large gains over diffusion and attention baselines. We show that this design has a systematic failure mode. The prior is fitted *per variable*, so it carries nothing across channels; when missingness comes in a contiguous block rather than scattered points, a variable's own history is severed and the prior degenerates to a straight line through the gap. On three of six benchmarks the resulting imputer is beaten by a plain transformer by up to 47.7%: in that regime the structured prior does active damage. We introduce GAPFLOW, which conditions the source prior on the *observed channels* through a ridge-regularised Gaussian conditional, drives the probability path by a two-sided distance to the nearest observation, and refines the sample by self-conditioning. GAPFLOW attains the lowest MAE against nine baselines in 72 of 72 settings, spanning six datasets, four missingness mechanisms, two ratios and four window lengths. Its gains are mechanism-dependent: 4.7–45.5% under block missingness against $\approx 0\%$ under MCAR, with no exception across six datasets.

1 INTRODUCTION

Real-world multivariate time series are rarely fully observed. Sensors fail and links drop, and the gaps they leave are contiguous rather than scattered. Imputation quality then bounds what any downstream model can learn from the record (Che et al., 2018; Du et al., 2023).

Prior work spans deterministic reconstruction, recurrent architectures with time-decayed states, and generative models (Tashiro et al., 2021; Lopez Alcaraz & Strodthoff, 2023; Liu et al., 2023a; Chen et al., 2024; Wang et al., 2025). In all of them missingness enters as input-side conditioning. How the *geometry* of the generative transport should respond to which entries are known is a separate question, and the one we take up.

Flow matching is a natural place to ask that question (Lipman et al., 2023; Liu et al., 2023b; Tong et al., 2024): it learns a velocity field carrying a source distribution to the data along a deterministic path, and the source is a free design choice. Recent work replaces Gaussian noise with a *structured prior* that already agrees with the observed entries: interpolation between adjacent observations (Qian et al., 2024), a Gaussian-process posterior (Kollovieh et al., 2025), or per-step context summaries (Hu et al., 2024; Zhang et al., 2024). Transport spent on coordinates that are already known is capacity not spent on the ones that are not.

Every such prior shares a single construction: it is fitted **one variable at a time**, so the source value at a missing entry of channel k depends on channel k's observations alone. Under scattered dropout the flanking observations lie a step or two away and the interpolant is nearly exact. Under a contiguous block they can lie dozens of steps apart, and what the prior returns in between is a straight chord that says nothing about the interior. The information is not gone: it sits in the other channels, which are redundant in a sensor network and are typically observed while channel k is not.

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.

We propose **GapFlow** (**Gap**-adaptive **Flow** matching), a flow-matching imputer whose source prior and probability path are organised by the geometry of the gap rather than by elapsed time within a channel. It conditions the prior mean at a missing entry on the *observed channels of the same step*, gated by how much of that channel's variance they explain; it drives the path speed by the *two-sided* distance to the nearest observation; and it refines the sample over several rounds (Figure 1). Where channels are independent the gate closes and GAPFLOW reduces to the per-variable prior exactly.

Across 72 settings the benefit of a structured prior is conditional, and outside that condition it becomes a liability. On the three datasets whose channels are most redundant, an interpolation-prior flow matcher loses to the strongest baseline under block missingness by 9.4% to 47.7%. GAPFLOW reverses the sign on all six of those rows and attains the lowest MAE in 72 of 72 settings against nine baselines. Its gains run 4.7% to 45.5% under block missingness against $\approx 0\%$ under MCAR on all six datasets, and the same contrast reappears when each component is measured separately.

Our primary contributions are:

- A **failure mode of structured-prior flow matching**: under contiguous-gap missingness in channel-redundant series it loses to a plain transformer by up to 47.7%, on three of six datasets.
- GAPFLOW, which adds a **cross-variable conditional prior**, a **two-sided-distance probability path** and **self-conditioned refinement**.
- **Mechanism-dependence**, the signature the account predicts: block gains of 4.7% to 45.5% against $\approx 0\%$ under MCAR on six of six datasets, and the same contrast at the component level ($94\times$ for the cross-variable prior).
- **State-of-the-art accuracy**, the lowest MAE in 72 of 72 settings against nine baselines, over six datasets, four mechanisms, two ratios and four window lengths, with the full grid, the negative results and the failed hypotheses reported in the appendix.

2 RELATED WORK

Time-series modelling under missingness and irregular sampling. Recurrent architectures handle gaps using time-decayed states (Che et al., 2018) or bidirectional recursion with consistency losses (Cao et al., 2018). Continuous-time approaches use neural ODEs (Rubanova et al., 2019) or controlled differential equations (Kidger et al., 2020), while attention-based models handle irregular sampling (Shukla & Marlin, 2021). Purely attentional imputers such as SAITS (Du et al., 2023) remain strong non-generative baselines and are the strongest baselines in several of our settings.

Probabilistic and generative models for incomplete time series. GP-VAE uses Gaussian-process latent priors (Fortuin et al., 2020; Titsias, 2009), while conditional diffusion performs imputation by denoising missing values conditioned on observations (Tashiro et al., 2021). Extensions combine diffusion with structured state-space models (Lopez Alcaraz & Strodthoff, 2023) or spatiotemporal conditioning (Liu et al., 2023a). Diffusion has also been studied for time-series anomaly detection (Pintilie et al., 2023), while Schrödinger bridges cast probabilistic imputation as a transport problem conditioned on observed values (Chen et al., 2023). These approaches differ in how temporal structure and observed context enter the generative process.

Flow matching and transport-based generative modelling. Flow matching learns a velocity field along a prescribed probability path (Lipman et al., 2023; Liu et al., 2023b; Tong et al., 2024). Time-series methods make substantially different choices for the transport construction. CLWF conditions transport from noise on observations (Qian et al., 2024), TSFlow uses Gaussian-process source priors (Kollovieh et al., 2025), and FlowTS adapts rectified flow to time-series generation (Hu et al., 2024). Other work considers trajectory flow matching (Zhang et al., 2024), optimal-transport or gradient-flow objectives (Wang et al., 2025; Chen et al., 2024), and conditional, mask-aware, or bridge-based formulations (Simkus & Gutmann, 2025; Liu et al., 2025; Xu et al., 2025; Park et al., 2026).

These methods differ in whether observed information enters through conditioning, the source distribution, or the transport objective. Our focus is narrower: we isolate a strong per-variable interpolation source, FM-INTERP, whose missing source values depend only on observations from the same channel. This construction works well for short gaps but cannot exploit informative peer

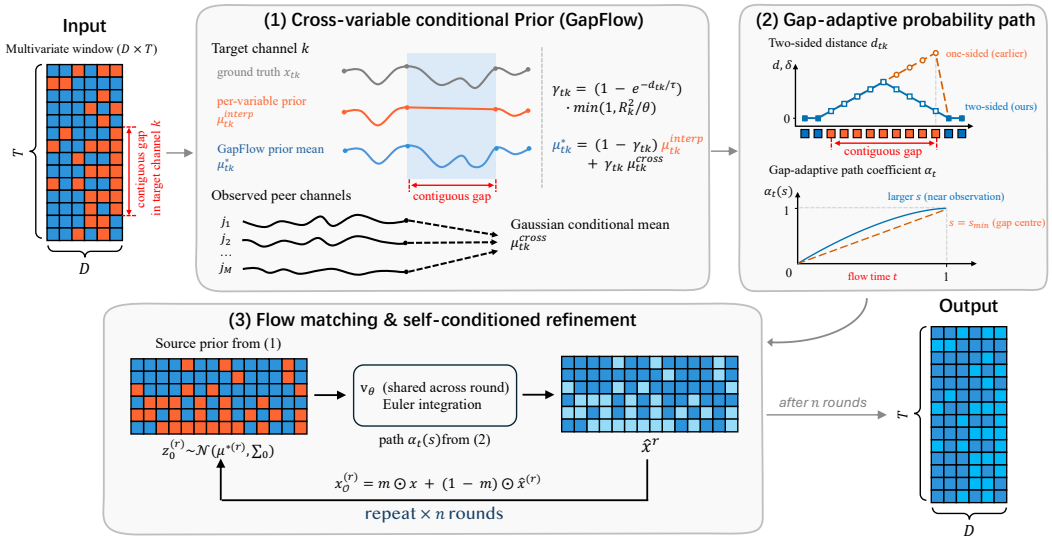

Figure 1: Overview of GapFlow. **(1)** Inside a contiguous gap of channel k the per-variable interpolant returns a straight chord (orange) that departs from the truth, while the peer channels $j_1, \ldots, j_M$ are observed at the same step; a ridge-regularised Gaussian conditional turns them into μ^{cross}, blended by γ, which closes when they explain none of channel k's variance. **(2)** The two-sided distance d peaks at gap centres and sets the path speed, so at any t an entry near an observation has travelled further towards the data than one at a gap centre. **(3)** Sampling repeats three times, feeding each estimate back into the prior as pseudo-observations; the mask is never modified.

channels inside long contiguous gaps. GapFlow addresses this bottleneck by placing cross-variable information directly in the source prior.

3 Method

3.1 Preliminaries

Let $x \in \mathbb{R}^{T \times D}$ be a window of D channels over T steps and $m \in \{0,1\}^{T \times D}$ the observation mask, with $m_{tk} = 1$ when x_{tk} is observed. Imputation seeks $\hat{x}_{tk}$ at every $m_{tk} = 0$. Throughout, a subscript pair tk indexes a step and a channel, while a bare subscript t on x_t or α_t is the flow time in $[0,1]$. Conditional flow matching learns a velocity field v_θ transporting a source $z_0 \sim p_0$ to the data at $t{=}1$ along the interpolant $x_t = (1 - \alpha_t)z_0 + \alpha_t x$, by regressing v_θ onto the conditional target $u_t = \dot{\alpha}_t\,(x - z_0)$, which reduces to $x - z_0$ under the usual linear $\alpha_t = t$:

$$\mathcal{L}_{\text{CFM}} = \mathbb{E}_{t,\,x,\,z_0}\left[\|v_\theta(x_t, t, c) - u_t\|_2^2\right], \qquad t \sim \mathcal{U}(0,1), \tag{1}$$

conditioned on $c = (x \odot m,\ m,\ g)$, where g encodes the missingness geometry. At inference we integrate from z_0 and return $\hat{x} = m \odot x + (1 - m) \odot z_1$. Because (1) is defined over all coordinates, a Gaussian source makes the network transport the observed entries as well, which is what a structured p_0 is meant to avoid.

3.2 The per-variable bottleneck

A structured source prior sets $p_0 = \mathcal{N}(\mu^{\text{interp}}, \Sigma_0)$ with

$$\mu_{tk}^{\text{interp}} = \frac{(t_k^+ - t)\,x_{t_k^- k} + (t - t_k^-)\,x_{t_k^+ k}}{t_k^+ - t_k^-}, \qquad \sigma_{tk} = \sigma_{\text{obs}}\,m_{tk} + \sigma_{\max}(1 - m_{tk}), \tag{2}$$

where t_k^-, t_k^+ are the observed steps of channel k bracketing t. Two properties follow. First, μ_{tk}^{interp} depends on channel k alone. Second, as the bracketing observations move apart it tends to a straight chord whose error grows with the gap length. Writing δ_{tk} for the steps elapsed since the last

observation of channel k, under MCAR at rate r the expected gap is $1/(1-r)$ steps and the chord is accurate; under block missingness a gap spans rT steps and the chord is uninformative.

The lost information is recoverable from other channels whenever they are redundant. We quantify redundancy by the **effective rank** of the channel covariance, $\mathrm{er} = (\sum_i \lambda_i)^2 / \sum_i \lambda_i^2$, the participation ratio of its eigenvalue spectrum. It equals D when channels are uncorrelated and approaches 1 when a single component explains the variance; across our six datasets it spans 1.17 to 7.99. On the lowest-rank datasets the mean absolute channel correlation exceeds 0.75, so a channel that is observed during another channel's gap is strongly informative about the missing values.

A per-variable prior is blind to this by construction, and the failure is not silent: the chord does not abstain in the middle of a long gap, it asserts a confidently wrong value, and (1) then trains the network to transport away from it. Capacity a Gaussian source would have spent on the missing coordinates goes instead into undoing a bad initialisation, which is why the failure appears as an inversion rather than a shrinking margin: on the three lowest-rank datasets FM-INTERP falls behind a plain attention imputer.

3.3 CROSS-VARIABLE CONDITIONAL PRIOR

We replace the per-variable mean at missing entries by a Gaussian conditional on the observed channels of the same step. Let $\Sigma \in \mathbb{R}^{D\times D}$ be the channel covariance estimated on the training split with pairwise-complete counts, and at step t partition channels into observed $\mathcal{O}$ and missing $\mathcal{M}$:

$$\mu_{\mathcal{M}}^{\mathrm{cross}} \;=\; \Sigma_{\mathcal{M}\mathcal{O}}\big(\Sigma_{\mathcal{O}\mathcal{O}} + \lambda I\big)^{-1} x_{\mathcal{O}}, \tag{3}$$

with ridge λ for numerical stability. The covariance is formed once per training run, with a channel pair contributing wherever both are observed rather than only on complete rows. Under our training missingness schedule this recovers between 38 and 46 times as many usable pairs as listwise deletion, measured across the four twenty-channel datasets. Because $\mathcal{O}$ and $\mathcal{M}$ change from step to step, (3) is solved as a single batched linear system over all steps of all windows rather than grouped by pattern, which costs a fraction of one integration (Appendix E).

Applying (3) everywhere would be a mistake: at short gaps the interpolant is already close to exact and a regression across channels adds variance for no bias reduction, and on independent channels the regression is uninformative at any gap length. We therefore blend (3) with (2) by a gap-length weight, gated by how predictable the target channel is:

$$\mu^{\star}_{tk} = \big(1-\gamma_{tk}\big)\,\mu^{\mathrm{interp}}_{tk} + \gamma_{tk}\,\mu^{\mathrm{cross}}_{tk}, \qquad \gamma_{tk} = \underbrace{\big(1-e^{-d_{tk}/\tau}\big)}_{\text{gap-length weight}} \cdot \underbrace{\min\!\big(1,\; R^2_k/\theta\big)}_{\text{predictability gate}}, \tag{4}$$

where $R^2_k = 1 - \Sigma^{|\mathcal{O}}_{kk}/\Sigma_{kk}$ is the fraction of channel k's variance explained by the observed channels. On a dataset with independent channels $R^2 \approx 0$, the gate closes, and (4) recovers the per-variable prior exactly.

3.4 TWO-SIDED DISTANCE AND THE PROBABILITY PATH

The elapsed time δ used by earlier designs (Che et al., 2018) is one-sided: it grows monotonically through a gap and is maximal just before the gap ends, where the *next* observation is one step away. We use

$$d_{tk} \;=\; \min\!\big(\delta^{\mathrm{back}}_{tk},\; \delta^{\mathrm{fwd}}_{tk}\big), \tag{5}$$

the distance to the nearest observation in either direction, which is zero at observed entries and peaks at gap centres. The distinction matters most where our method matters: under Block missingness δ overestimates d by 80% on average and the two correlate at only 0.22, whereas under MCAR the overestimate falls to 27% and the correlation rises to 0.40. Substituting one for the other is therefore not a reparameterisation, and it does most damage in the contiguous-gap regime. Each entry then follows its own schedule, indexed by a speed that its distance sets:

$$\alpha_t(s) \;=\; 1-(1-t)^{\,s}, \qquad s_{tk} \;=\; s_{\min} + (s_{\mathrm{near}} - s_{\min})\,e^{-d_{tk}/\tau_s}, \tag{6}$$

at missing entries, and $s_{tk} = s_{\mathrm{obs}}$ where the entry is observed. Every speed is at least 1, which keeps α_t increasing and C^1 in t, so the target of (1) becomes $u_t = s_{tk}(1-t)^{s_{tk}-1}(x - z_0)$ entrywise and

Table 1: Imputation results (MAE ↓, T=48). Best in **bold**, second best underlined. GAPFLOW wins **48/48** settings against the nine published baselines. FM-INTERP is not one of them: it is the per-variable interpolation prior of (2), built by us, and is column (A) of Table 2. All methods are trained by us for 500 epochs under identical splits and masks.

			FLOW-MATCHING						DETERMINISTIC / GENERATIVE				
Dataset	Mechanism	Rate	CLWF	FlowTS	FM-Interp	PSW-I	TG-MSFM	TSFlow	SAITS	BRITS	CSDI	GP-VAE	**Ours**
Synthetic	MCAR	30%	0.447	0.681	0.100	1.313	0.341	0.112	0.118	0.148	0.133	0.768	**0.099**
	MCAR	50%	0.667	0.789	**0.110**	1.689	0.449	0.138	0.144	0.195	0.161	0.799	0.112
	Block	30%	0.697	0.781	0.368	1.768	0.631	0.555	0.499	0.864	0.406	0.778	**0.338**
	Block	50%	0.748	0.815	0.468	1.276	0.680	0.644	0.578	0.849	0.502	0.784	**0.446**
	MNAR	30%	0.620	0.861	0.112	1.423	0.468	0.119	0.127	0.174	0.131	0.994	**0.107**
	MNAR	50%	0.828	0.915	0.129	1.725	0.582	0.141	0.181	0.240	0.162	0.924	**0.118**
	Irregular	30%	0.795	0.845	0.205	1.433	0.582	0.248	0.347	0.361	0.275	0.801	**0.199**
	Irregular	50%	0.802	0.846	0.202	1.774	0.582	0.245	0.344	0.359	0.227	0.800	**0.199**
ETTm1	MCAR	30%	0.146	0.370	0.076	1.846	0.183	0.118	0.082	0.105	0.112	0.406	**0.074**
	MCAR	50%	0.181	0.412	0.086	2.280	0.230	0.129	0.094	0.123	0.137	0.555	**0.084**
	Block	30%	0.160	0.395	0.107	1.701	0.270	0.245	0.127	0.154	0.190	0.491	**0.096**
	Block	50%	0.202	0.440	0.147	1.733	0.364	0.268	0.174	0.249	0.223	0.565	**0.126**
	MNAR	30%	0.302	0.427	0.116	2.321	0.257	0.185	0.120	0.179	0.186	0.697	**0.108**
	MNAR	50%	0.324	0.473	0.159	2.467	0.343	0.241	0.178	0.263	0.212	0.788	**0.140**
	Irregular	30%	0.317	0.435	0.102	2.396	0.312	0.142	0.124	0.152	0.190	0.734	**0.099**
	Irregular	50%	0.281	0.432	0.102	2.543	0.316	0.138	0.117	0.149	0.199	0.735	**0.101**
Traffic	MCAR	30%	0.222	0.239	**0.113**	0.364	0.238	0.195	0.144	0.148	0.181	0.331	**0.113**
	MCAR	50%	0.239	0.289	0.136	0.502	0.264	0.248	0.167	0.177	0.254	0.357	**0.134**
	Block	30%	0.268	0.288	0.249	0.480	0.311	0.476	0.201	0.215	0.251	0.372	**0.170**
	Block	50%	0.284	0.368	0.295	0.523	0.349	0.496	0.227	0.248	0.308	0.411	**0.203**
	MNAR	30%	0.600	0.522	0.235	0.463	0.465	0.467	0.324	0.319	0.379	0.753	**0.211**
	MNAR	50%	0.674	0.748	0.423	0.558	0.665	0.568	0.544	0.526	0.480	0.838	**0.348**
	Irregular	30%	0.376	0.404	0.169	0.658	0.291	0.364	0.202	0.211	0.296	0.402	**0.162**
	Irregular	50%	0.320	0.413	0.168	0.723	0.291	0.318	0.203	0.219	0.336	0.445	**0.161**
PEMS-BAY	MCAR	30%	0.133	0.221	**0.058**	0.591	0.155	0.070	0.079	0.070	0.076	0.206	**0.058**
	MCAR	50%	0.137	0.253	0.065	0.747	0.174	0.082	0.092	0.085	0.094	0.238	**0.064**
	Block	30%	0.197	0.241	0.124	0.539	0.243	0.224	0.153	0.149	0.153	0.237	**0.107**
	Block	50%	0.190	0.268	0.150	0.609	0.320	0.221	0.161	0.176	0.202	0.251	**0.121**
	MNAR	30%	0.242	0.203	0.086	0.588	0.201	0.103	0.107	0.103	0.102	0.298	**0.083**
	MNAR	50%	0.228	0.232	0.109	0.684	0.228	0.142	0.128	0.132	0.144	0.328	**0.107**
	Irregular	30%	0.301	0.312	0.077	0.795	0.195	0.091	0.116	0.114	0.122	0.266	**0.076**
	Irregular	50%	0.219	0.323	**0.076**	0.870	0.197	0.090	0.114	0.117	0.131	0.271	**0.076**
Electricity	MCAR	30%	0.202	0.329	0.094	0.474	0.189	0.148	0.122	0.114	0.139	0.327	**0.090**
	MCAR	50%	0.197	0.344	0.103	0.651	0.212	0.176	0.135	0.133	0.160	0.346	**0.097**
	Block	30%	0.225	0.352	0.226	0.552	0.239	0.354	0.165	0.170	0.206	0.330	**0.133**
	Block	50%	0.233	0.373	0.260	0.674	0.298	0.337	0.176	0.194	0.249	0.411	**0.142**
	MNAR	30%	0.231	0.304	0.108	0.466	0.248	0.186	0.134	0.131	0.152	0.486	**0.100**
	MNAR	50%	0.280	0.394	0.136	0.573	0.298	0.237	0.170	0.178	0.185	0.569	**0.118**
	Irregular	30%	0.345	0.406	0.121	0.736	0.241	0.249	0.165	0.169	0.191	0.413	**0.113**
	Irregular	50%	0.275	0.420	0.121	0.800	0.242	0.223	0.163	0.170	0.205	0.399	**0.113**
Solar	MCAR	30%	0.092	0.240	0.040	0.208	0.145	0.043	0.062	0.050	0.060	0.138	**0.039**
	MCAR	50%	0.098	0.262	0.048	0.279	0.166	0.051	0.073	0.071	0.077	0.154	**0.047**
	Block	30%	0.112	0.245	0.092	0.224	0.251	0.160	0.089	0.084	0.151	0.155	**0.076**
	Block	50%	0.111	0.240	0.109	0.268	0.398	0.163	0.089	0.091	0.131	0.212	**0.080**
	MNAR	30%	0.143	0.203	0.053	0.208	0.169	0.057	0.076	0.072	0.070	0.216	**0.050**
	MNAR	50%	0.147	0.224	0.067	0.266	0.197	0.074	0.085	0.096	0.097	0.241	**0.063**
	Irregular	30%	0.292	0.279	0.058	0.321	0.190	0.060	0.087	0.088	0.097	0.204	**0.055**
	Irregular	50%	0.181	0.280	0.059	0.348	0.191	0.059	0.089	0.088	0.123	0.200	**0.056**

the sampler still steps uniformly in t. Since $\alpha_t(s)$ is increasing in s, at any t an entry close to an observation has already travelled further towards the data than one at a gap centre, which stays near the source for longer. What the prior already gets right is committed early; what it cannot know keeps room for the network to act on.

3.5 SELF-CONDITIONING

Equation (3) conditions on the observed channels at a step, but after one integration we also hold an *estimate* of the missing ones. We sample n times; round $r > 1$ recomputes (3) with the previous round's output substituted into $x_{\mathcal{O}}$ at the missing coordinates,

$$x_{\mathcal{O}}^{(r)} \;=\; m \odot x \;+\; (1-m) \odot \hat{x}^{(r-1)}, \tag{7}$$

then re-integrates. The mask is never modified: pseudo-observations enter the prior only, so $\mathcal{L}_{\mathrm{CFM}}$ and the reported MAE remain defined on the true missing set. We use n=3; the accuracy–cost trade-off is in Section 5.4 and Appendix I.

4 EXPERIMENTAL SETUP

Datasets and evaluation protocol. We evaluate on six datasets spanning four domains, ordered by channel effective rank: **Synthetic** (D=8, er 7.99; sinusoidal), **ETTm1** (D=7, 4.10; transformer temperatures) (Zhou et al., 2021), **Traffic** (D=20, 2.84; freeway occupancy) (Lai et al., 2018), **PEMS-BAY** (D=20, 1.55; freeway speeds) (Li et al., 2018), **Electricity** (D=20, 1.50; UCI load) (Dua & Graff, 2017) and **Solar** (D=20, 1.17; photovoltaic generation) (Lai et al., 2018); a seventh is deferred to Appendix B because its pre-standardised source breaks the channel-selection rule. The default length is T=48, and $T \in \{24, 96, 128\}$ on ETTm1 as well. Missingness is imposed under four mechanisms at two ratios (30%, 50%): **MCAR**, **Block** (contiguous per-feature gaps), **MNAR** (value-dependent dropout) and **Irregular** (asynchronous per-feature patterns), giving 48 settings at T=48 and 72 in total. Channels are z-score normalised per variable over the full series, a leak that Appendix A controls for, and data is split 80/20 with seed 42; no validation split is held out, since early stopping is disabled everywhere. Evaluation masks are bit-identical across methods, verified by digest comparison over all 48 combinations. Full specifications, including a caveat on the rate the Irregular generator actually realises, are in Appendix A.

Baselines and metrics. We compare against nine methods under a unified protocol: **SAITS** (Du et al., 2023) and **BRITS** (Cao et al., 2018) (deterministic); **CSDI** (Tashiro et al., 2021) and **GP-VAE** (Fortuin et al., 2020) (diffusion/VAE); **FlowTS** (Hu et al., 2024), **CLWF** (Qian et al., 2024), **TSFlow** (Kollovieh et al., 2025), **PSW-I** (Wang et al., 2025) and **TG-MSFM** (Wang & Sugiyama, 2026) (flow-matching). We additionally report FM-INTERP, the interpolation-prior design of (2), which is our comparison starting point and column (A) of Table 2. All are run by us for 500 epochs with early stopping disabled, using the PyPOTS implementations (Du, 2023) where available. Two asymmetries *favour* the baselines and we retain them: their training missingness matches the evaluation mechanism, and their harness selects checkpoints on the evaluation split whereas ours selects on the training split. We report **MAE** on held-out missing positions; Appendix G adds four probabilistic metrics, on two of which GAPFLOW is worse than FM-INTERP.

Every flow-matching arm, ours and the reimplemented baselines alike, uses the same velocity network, a 4-layer Transformer encoder (d_{model}=256, 4 heads, ∼3.9M parameters), so capacity is not a confound within that family; the PyPOTS baselines keep their published architectures, which are seven to four hundred times smaller. Inference uses Euler integration with 20 steps and averages K=20 samples per window, the protocol the baselines were evaluated under, and GAPFLOW and FM-INTERP are run with three seeds in every setting. Every entry is the mean over the available runs; the run-to-run spread is in Appendix K. Training missingness is drawn per sample from a random-rate MCAR schedule independent of the evaluation mechanism, so one trained model is evaluated under all eight mask configurations of its dataset. Full details are in Appendices D and C.

5 RESULTS AND ANALYSIS

5.1 MAIN IMPUTATION RESULTS

GAPFLOW attains the lowest MAE in **72 of 72** settings against the nine published baselines (Table 1 shows the 48 settings at T=48 and Table 3 the remaining 24). The median margin over the strongest baseline in each row is 16.4% and never falls below 3.2%. Against FM-INTERP, our own starting point rather than a published method, GAPFLOW is better in 70 of 72. Both exceptions are MCAR settings, where the interpolant is already near-optimal and there is nothing for cross-variable information to repair: Synthetic at 50% by 1.5% and Traffic at 30% by 0.2%.

Which method comes second is itself informative. The strongest published baseline is the attention imputer in 22 of the 48 rows, the GP-prior flow matcher in 14, the recurrent imputer in 7 and the diffusion imputer in 5: the deterministic and the flow-matching families split second place roughly evenly, so the framework's advantage over older designs is conditional rather than uniform. Identifying that condition occupies the rest of this section.

Where the structured prior fails. The per-variable interpolation prior is a strong construction: counting it alongside the published methods, it is the runner-up in 41 of the 48 rows, behind only GAPFLOW. A design that places second almost everywhere is not a weak baseline, which is what

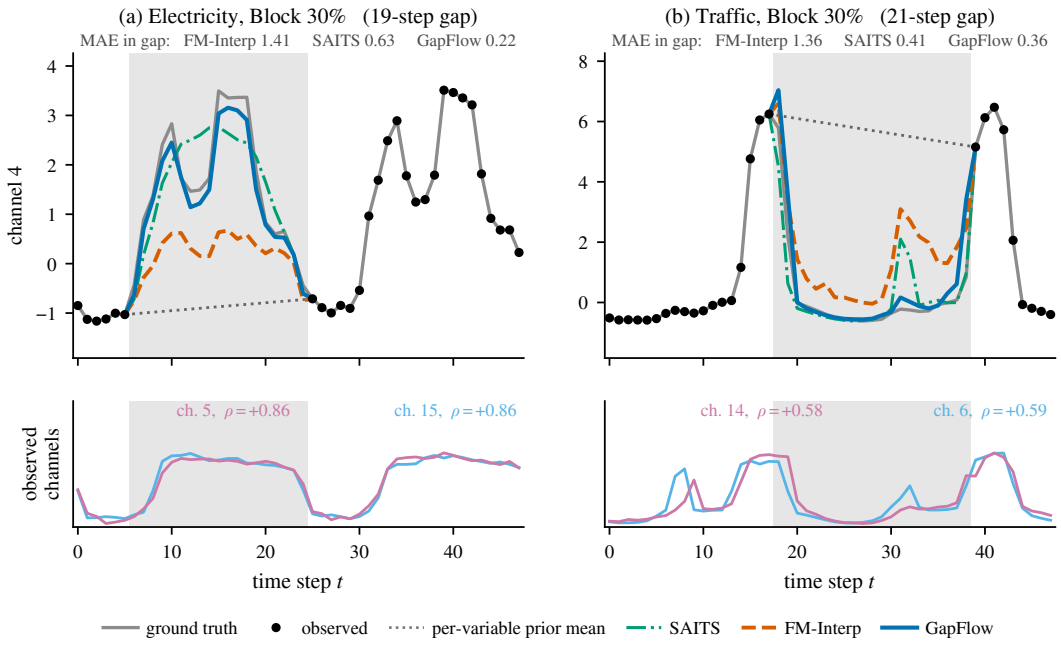

Figure 2: Imputation across a contiguous gap at 30% Block missingness on **(a)** Electricity and **(b)** Traffic. The dotted chord is the per-variable prior mean; inside the gap it is wrong, and FM-INTERP is pulled towards it. Below are the correlated channels that stay observed across the same span. Each window is the gap whose truth departs furthest from its own chord, selected from truth and mask alone; over all 1386 and 914 such gaps GAPFLOW is ahead in 84% and 77% (Appendix J).

makes the rows where it inverts worth reading. On the three most channel-redundant datasets it loses to the strongest baseline in each row, and loses by a wide margin: Electricity +37.2% and +47.7% at 30% and 50% Block missingness, Traffic +23.8% and +30.2%, Solar +9.4% and +22.4%. A structured prior is supposed to help; on these rows it hurts. GAPFLOW reverses the sign on all six, reaching −19.2% and −19.6%, −15.8% and −10.4%, and −9.2% and −10.3% against the same baseline. Figure 2 shows what the failure looks like inside a single window. Those three datasets carry the three lowest channel effective ranks in the study, 1.50, 2.84 and 1.17, so each channel is largely predictable from the others—exactly the structure a per-variable prior cannot reach. Where effective rank is higher the same prior does not fail: on ETTm1 (4.10) and Synthetic (7.99) FM-INTERP beats every baseline on the Block rows. The failure is a property neither of Block missingness alone nor of channel redundancy alone, but of their conjunction.

Gains are mechanism-dependent, without exception. If the account of Section 3.2 is right, gains should concentrate where gaps are contiguous and vanish where they are not. They do, on **six of six** datasets: Block gains run from 4.7% on Synthetic to 45.5% on Electricity, the corresponding MCAR gains from −1.5% to 5.2%. Extra capacity or a better optimiser would not produce this pattern; both would lift the two mechanisms together. Nor is the contrast confined to MAE: the energy score, which scores the whole window as a vector, improves by 9.1% under Block and degrades by 5.7% under MCAR (Appendix G).

5.2 ABLATION: WHAT DRIVES THE GAINS?

Table 2 adds one component at a time and reports, under each entry, its change relative to the configuration on its left. Read column by column, the ladder tells a consistent story: the same mechanism-dependence that separates the Block rows from the MCAR rows in Table 1 reappears *inside* each component. Because the components are measured separately, this is independent evidence for the account of Section 3.2 rather than a restatement of it.

The cross-variable prior of (3) is the dominant step. It accounts for a 13.0% median reduction across the twelve Block settings against 0.14% across the twelve MCAR settings, a ratio of 94×, and it is

Table 2: Incremental ablation (MAE ↓, 4 d.p.). Configuration (A) is the per-variable interpolation prior of (2), the standard structured-prior construction; (B)–(E) add one component at a time; the small figure under each entry is its relative change against the configuration on its left (blue = improvement, orange = regression). Best per row in **bold**. Δ: gain of (E) over (A). The final two rows give each component's median change over all 12 Block and 12 MCAR settings, including the rates not shown above.

Dataset	Setting	(A) Per-variable prior	(B) +Cross-var. prior	(C) +Gap-adapt. path	(D) +Self-conditioning	(E) +R^2 hinge (GapFlow)	Δ
Synthetic	MCAR 30%	0.0996	0.0995 (-0.1)	0.0997 (+0.2)	0.1005 (+0.8)	**0.0994** (-1.1)	−0%
	Block 30%	0.3681	0.3450 (-6.3)	0.3384 (-1.9)	**0.3196** (-5.5)	0.3382 (+5.8)	−8%
	Block 50%	0.4677	0.4418 (-5.5)	0.4359 (-1.3)	**0.4146** (-4.9)	0.4457 (+7.5)	−5%
ETTm1	MCAR 30%	0.0760	0.0760 (-0.1)	0.0753 (-0.8)	0.0742 (-1.5)	**0.0735** (-0.9)	−3%
	Block 30%	0.1066	0.1144 (+7.3)	0.1098 (-4.0)	0.0971 (-11.6)	**0.0965** (-0.6)	−10%
	Block 50%	0.1474	0.1605 (+8.9)	0.1547 (-3.6)	**0.1258** (-18.7)	0.1262 (+0.4)	−14%
Traffic	MCAR 30%	0.1125	0.1122 (-0.3)	**0.1106** (-1.5)	0.1128 (+2.0)	0.1127 (-0.1)	+0%
	Block 30%	0.2493	0.1876 (-24.8)	0.1831 (-2.4)	**0.1695** (-7.4)	0.1697 (+0.1)	−32%
	Block 50%	0.2954	0.2186 (-26.0)	0.2126 (-2.8)	**0.2029** (-4.6)	0.2033 (+0.2)	−31%
PEMS-BAY	MCAR 30%	0.0580	0.0581 (+0.2)	0.0580 (-0.2)	0.0577 (-0.6)	**0.0576** (-0.1)	−1%
	Block 30%	0.1243	0.1163 (-6.5)	0.1149 (-1.2)	**0.1067** (-7.1)	0.1068 (+0.0)	−14%
	Block 50%	0.1500	0.1308 (-12.8)	0.1289 (-1.4)	**0.1205** (-6.5)	0.1215 (+0.8)	−19%
Electricity	MCAR 30%	0.0935	0.0931 (-0.5)	0.0912 (-2.1)	0.0902 (-1.1)	**0.0901** (-0.1)	−4%
	Block 30%	0.2259	0.1508 (-33.2)	0.1455 (-3.5)	0.1331 (-8.5)	**0.1331** (-0.0)	−41%
	Block 50%	0.2602	0.1606 (-38.3)	0.1538 (-4.2)	0.1420 (-7.6)	**0.1418** (-0.2)	−46%
Solar	MCAR 30%	0.0399	0.0397 (-0.5)	**0.0384** (-3.2)	0.0387 (+0.8)	0.0387 (-0.1)	−3%
	Block 30%	0.0916	0.0795 (-13.2)	0.0783 (-1.6)	**0.0760** (-2.9)	0.0761 (+0.1)	−17%
	Block 50%	0.1087	0.0849 (-21.9)	0.0828 (-2.4)	**0.0796** (-3.9)	0.0797 (+0.1)	−27%
median Δ over all Block settings		–	**-13.0%**	**-2.4%**	**-6.8%**	**+0.1%**	
median Δ over all MCAR settings		–	-0.14%	-1.2%	+0.3%	-0.1%	

where the three datasets that fail under a per-variable prior recover most of their deficit: -33.2% on Electricity Block 30% and -24.8% on Traffic Block 30% from this single change. The gap-adaptive path of (6) is the smallest of the three in magnitude, at 2.4% on Block against 1.2% on MCAR, but it is the most consistent: it helps in 12 of the 12 Block settings, whereas the other components each regress somewhere. Self-conditioning contributes 6.8% on Block against a 0.3% regression on MCAR, and its profile across mechanisms is the most uneven of the three, helping strongly under MNAR and hurting under Irregular sampling (Appendix H).

Two entries in the ladder run the wrong way. On Synthetic—the one dataset whose channels are close to independent—the cross-variable prior gives back most of what the later components gain, which is the behaviour the predictability gate is designed to produce and the reason GapFlow improves on FM-Interp by only 4.7% and 8.1% on its two Block settings. The R^2 hinge is neutral on Block ($+0.1\%$) and mildly harmful under MNAR; it earns its place through MCAR and Irregular, where it helps in 23 of 24 settings. An observation-rate safeguard that disables self-conditioning below a threshold never fires on any dataset reported here, since observation rates run 50–70% against a threshold of 0.3; it contributes exactly zero to every number in this paper.

5.3 Sequence Length Generalisation

The window length fixes how long a Block gap is at a given ratio, so it is the axis on which the account should show most clearly. Table 3 runs ETTm1 at $T \in \{24, 96, 128\}$ against the same nine baselines. GapFlow is best in all 24, and the Block gain tracks the gap it has to cross: -0.8%, -9.5%, -11.6% and -8.2% at $T = 24, 48, 96, 128$ for 30% missingness. At T=24 a 30% Block gap is seven steps, short enough for a chord to remain a fair interpolant, and the gain is close to nothing; by T=96 it is 29 steps and the gain is an order of magnitude larger. The fall-back at T=128 carries a confound we cannot separate: window count drops inversely with T, from 1451 to 544, so the covariance estimate and the network both see less data (Appendix F).

Table 3: Sequence-length generalisation on ETTm1 (MAE ↓), with the same columns as Table 1. GAPFLOW wins **24/24** settings; the strongest published baseline is SAITS in 23 of them. With the 48 settings of Table 1 these are the 72 reported in the text. Best in **bold**, second best underlined.

			FLOW-MATCHING						DETERMINISTIC / GENERATIVE				
Length	Mechanism	Rate	CLWF	FlowTS	FM-Interp	PSW-I	TG-MSFM	TSFlow	SAITS	BRITS	CSDI	GP-VAE	**Ours**
T=24	MCAR	30%	0.127	0.326	0.078	1.627	0.194	0.125	0.084	0.122	0.114	0.416	**0.077**
	MCAR	50%	0.154	0.347	0.090	2.027	0.257	0.139	0.100	0.152	0.141	0.578	**0.088**
	Block	30%	0.189	0.369	0.119	1.531	0.368	0.238	0.138	0.252	0.187	0.545	**0.118**
	Block	50%	0.185	0.390	0.145	1.537	0.435	0.257	0.196	0.365	0.237	0.599	**0.132**
	MNAR	30%	0.210	0.314	0.117	2.020	0.263	0.176	0.113	0.198	0.165	0.605	**0.110**
	MNAR	50%	0.231	0.342	0.135	2.109	0.324	0.188	0.145	0.246	0.184	0.727	**0.125**
	Irregular	30%	0.255	0.366	0.108	2.175	0.368	0.151	0.123	0.180	0.171	0.750	**0.106**
	Irregular	50%	0.215	0.360	0.109	2.335	0.376	0.148	0.126	0.177	0.190	0.749	**0.108**
T=96	MCAR	30%	0.177	0.418	0.079	1.461	0.191	0.125	0.085	0.096	0.123	0.415	**0.077**
	MCAR	50%	0.232	0.484	0.090	1.847	0.228	0.136	0.098	0.112	0.177	0.582	**0.087**
	Block	30%	0.217	0.476	0.125	1.503	0.270	0.273	0.144	0.157	0.210	0.513	**0.111**
	Block	50%	0.264	0.550	0.158	1.470	0.357	0.299	0.173	0.259	0.261	0.576	**0.140**
	MNAR	30%	0.372	0.481	0.122	1.819	0.276	0.198	0.127	0.157	0.192	0.811	**0.113**
	MNAR	50%	0.415	0.537	0.180	2.016	0.371	0.288	0.205	0.230	0.237	0.832	**0.160**
	Irregular	30%	0.386	0.501	0.109	2.031	0.294	0.151	0.128	0.137	0.198	0.730	**0.107**
	Irregular	50%	0.350	0.508	0.110	2.250	0.299	0.145	0.122	0.136	0.202	0.733	**0.108**
T=128	MCAR	30%	0.196	0.481	0.082	1.484	0.224	0.125	0.089	0.100	0.133	0.441	**0.081**
	MCAR	50%	0.251	0.547	0.092	1.850	0.270	0.135	0.107	0.113	0.172	0.617	**0.090**
	Block	30%	0.237	0.525	0.127	1.565	0.295	0.269	0.158	0.163	0.230	0.540	**0.117**
	Block	50%	0.324	0.593	0.175	1.568	0.419	0.310	0.207	0.233	0.315	0.619	**0.155**
	MNAR	30%	0.406	0.546	0.126	1.970	0.335	0.190	0.131	0.156	0.179	0.839	**0.117**
	MNAR	50%	0.443	0.613	0.205	1.998	0.472	0.315	0.231	0.238	0.249	0.871	**0.187**
	Irregular	30%	0.417	0.582	0.110	2.033	0.351	0.145	0.131	0.143	0.195	0.781	**0.108**
	Irregular	50%	0.373	0.579	0.110	2.204	0.356	0.142	0.130	0.142	0.275	0.782	**0.109**

5.4 GAP-DEPTH AND COST ANALYSIS

Row-level numbers establish that something goes wrong under Block missingness but not *where* inside a window, and a method could improve a Block row by imputing the easy edges of a gap while leaving its centre untouched. Binning every imputed position by its two-sided distance d separates them. Under Block missingness the relative gap between FM-INTERP and GAPFLOW widens monotonically, from 9.5% at $d \in [1, 2]$ to 18.7%, 26.2% and 32.7% beyond twelve steps, with GAPFLOW ahead in 12 of 12 dataset–ratio cells in the first three buckets. Under MCAR the same buckets separate by 1.5% and 2.9%, and the deepest are empty (Appendix J). A capacity account would have to explain why capacity is worth 1.5% under MCAR and 32.7% at $d > 12$ under Block, and why the boundary falls at the gap depth beyond which a chord stops being a fair interpolant rather than at some property of the datasets. A second measurement agrees: the ratio of missing- to observed-coordinate arc length along the integration path has median 18.2 for FM-INTERP and 16.4 for GAPFLOW: both priors concentrate transport where it is needed, and the reduction under GAPFLOW is what a source starting closer to the target should produce.

Inference costs 1200 network evaluations per window against 400 for FM-INTERP, a $3.0\times$ increase dominated not by the refinement but by the 20-sample averaging the evaluation protocol inherits: at one sample GAPFLOW needs 60 evaluations and still beats FM-INTERP at twenty on Block settings, at one sixth the cost (Appendix I).

6 CONCLUSION

Structured source priors turn flow matching into a strong imputer; the same per-variable construction makes it fail exactly where sensor data is most redundant and gaps are longest. We showed the failure is large—up to 47.7% worse than a plain transformer—characterised the regime where it occurs, and repaired it by conditioning the prior across channels and driving the path by a two-sided distance. GAPFLOW attains the lowest MAE in 72 of 72 settings, with gains concentrated in the contiguous-gap regime on all six datasets.

Two limitations bound the claim. Our gains require channel redundancy, but redundancy does not predict their size, as we tried and failed to show quantitatively (Appendix L). And the cross-variable conditional inherits any bias in the observed set, which is why the predictability gate is mildly harmful under value-dependent missingness; whether a mechanism-aware covariance estimate removes that penalty is the natural next question.

AI USE STATEMENT

Subtasks for which disclosure is required. We used generative AI tools for one: checking the manuscript's numerical claims against its own tables and the released artefacts, with every resulting correction confirmed by an author before it was made. We did not use them for any other required-disclosure subtask. The synthetic benchmark is produced by the deterministic sinusoidal generator of Appendix A rather than by a generative model, and every number, table and figure in this paper comes from experiments run by the authors.

Subtasks for which disclosure is recommended. We used generative AI tools to draft passages of the manuscript from results and conclusions supplied by the authors, to edit author-written text for readability, to write code, to produce figures, and to format and check bibliographic references. Beyond the subtasks listed above, no technical content originated with the tools, and no passage was retained without an author checking it against the underlying results.

We have reviewed all AI-assisted work. We take responsibility for the final content of this paper, including any text produced with the aid of generative AI tools.

REFERENCES

Wei Cao, Dong Wang, Jian Li, Hao Zhou, Lei Li, and Yitan Li. BRITS: Bidirectional recurrent imputation for time series. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

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

Song Chen. Beijing multi-site air quality. UCI Machine Learning Repository, 2017. URL https://archive.ics.uci.edu/dataset/501/beijing+multi+site+air+quality.

Yu Chen, Wei Deng, Shikai Fang, Fengpei Li, Nicole Tianjiao Yang, Yikai Zhang, Kashif Rasul, Shandian Zhe, Anderson Schneider, and Yuriy Nevmyvaka. Provably convergent schrödinger bridge with applications to probabilistic time series imputation. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 4485–4513. PMLR, 2023. URL https://proceedings.mlr.press/v202/chen23f.html.

Zhichao Chen, Haoxuan Li, Fangyikang Wang, Odin Zhang, Hu Xu, Xiaoyu Jiang, Zhihuan Song, and Hao Wang. Rethinking the diffusion models for missing data imputation: A gradient flow perspective. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-3558.

Wenjie Du. PyPOTS: A python toolbox for data mining on partially-observed time series. *arXiv preprint arXiv:2305.18811*, 2023. doi: 10.48550/arXiv.2305.18811. URL https://arxiv.org/abs/2305.18811.

Wenjie Du, David Côté, and Yan Liu. SAITS: Self-attention-based imputation for time series. *Expert Systems with Applications*, 219:119619, 2023. doi: 10.1016/j.eswa.2023.119619.

Dheeru Dua and Casey Graff. UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences, 2017. URL https://archive.ics.uci.edu/ml.

Vincent Fortuin, Dmitry Baranchuk, Gunnar Rätsch, and Stephan Mandt. GP-VAE: Deep probabilistic time series imputation. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 1651–1661. PMLR, 2020. URL https://proceedings.mlr.press/v108/fortuin20a.html.

Yang Hu, Xiao Wang, Zezhen Ding, Lirong Wu, Huatian Zhang, Stan Z. Li, Sheng Wang, Jiheng Zhang, Ziyun Li, and Tianlong Chen. FlowTS: Time series generation via rectified flow. *arXiv preprint arXiv:2411.07506*, 2024. doi: 10.48550/arXiv.2411.07506. URL https://arxiv.org/abs/2411.07506.

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

Marcel Kollovieh, Marten Lienen, David Lüdke, Leo Schwinn, and Stephan Günnemann. Flow matching with gaussian process priors for probabilistic time series forecasting. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=uxVBbSlKQ4.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long- and short-term temporal patterns with deep neural networks. In *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 95–104. Association for Computing Machinery, 2018. doi: 10.1145/3209978.3210006.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=SJiHXGWAZ.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Dengyi Liu, Honggang Wang, and Hua Fang. Impute-MACFM: Imputation based on mask-aware flow matching. *arXiv preprint arXiv:2509.23126*, 2025. doi: 10.48550/arXiv.2509.23126. URL https://arxiv.org/abs/2509.23126.

Mingzhe Liu, Han Huang, Hao Feng, Leilei Sun, Bowen Du, and Yanjie Fu. PriSTI: A conditional diffusion framework for spatiotemporal imputation. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pp. 1927–1939, 2023a. doi: 10.1109/ICDE55515.2023.00150.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=XVjTT1nw5z.

Juan Miguel Lopez Alcaraz and Nils Strodthoff. Diffusion-based time series imputation and forecasting with structured state space models. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=hHiIbk7ApW.

Jinseong Park, Seungyun Lee, Woojin Jeong, Yujin Choi, and Jaewook Lee. TimeBridge: Better diffusion prior design with bridge models for time series generation. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1135–1146. Association for Computing Machinery, 2026. doi: 10.1145/3770854.3780195.

Ioana Pintilie, Andrei Manolache, and Florin Brad. Time series anomaly detection using diffusion-based models. In *2023 IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 570–578, 2023. doi: 10.1109/ICDMW60847.2023.00080.

Weizhu Qian, Dalin Zhang, Yan Zhao, and Yunyao Cheng. Conditional lagrangian wasserstein flow for time series imputation. *arXiv preprint arXiv:2410.07550*, 2024. doi: 10.48550/arXiv.2410.07550. URL https://arxiv.org/abs/2410.07550.

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

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

Vaidotas Simkus and Michael U. Gutmann. CFMI: Flow matching for missing data imputation. *arXiv preprint arXiv:2506.09258*, 2025. doi: 10.48550/arXiv.2506.09258. URL https://arxiv.org/abs/2506.09258.

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

Michalis Titsias. Variational learning of inducing variables in sparse gaussian processes. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, pp. 567–574. PMLR, 2009. URL https://proceedings.mlr.press/v5/titsias09a.html.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=CD9Snc73AW.

Hangtian Wang and Mahito Sugiyama. Time-gated multi-scale flow matching for time-series imputation. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=txvc61ONbs.

Hao Wang, Zhengnan Li, Haoxuan Li, Xu Chen, Mingming Gong, Bin Chen, and Zhichao Chen. Optimal transport for time series imputation. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=xPTzjpIQNp.

Huibo Xu, Runlong Yu, Likang Wu, Xianquan Wang, and Qi Liu. Bridging the last mile of prediction: Enhancing time series forecasting with conditional guided flow matching. *arXiv preprint arXiv:2507.07192*, 2025. doi: 10.48550/arXiv.2507.07192. URL https://arxiv.org/abs/2507.07192.

Xi Zhang, Yuan Pu, Yuki Kawamura, Andrew Loza, Yoshua Bengio, Dennis L. Shung, and Alexander Tong. Trajectory flow matching with applications to clinical time series modelling. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-3404.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(12):11106–11115, 2021. doi: 10.1609/aaai.v35i12.17325.

Appendix Overview

A. Datasets and Missingness Details
B. The Excluded Seventh Dataset
C. Implementation Details
D. Baselines, Fairness, and Evaluation Metrics
E. Additional Method Details
F. Sequence Length Generalisation
G. Uncertainty Calibration (a negative result)
H. Ablation Details
I. Accuracy–Cost Trade-off
J. Extended Analysis (Transport Geometry, Qualitative Window, Gap-Bucketed)
K. Multi-Seed Evaluation
L. Pre-Registration
M. Negative and Neutral Design Choices
N. Reproducibility

A Datasets and Missingness Details

A.1 Benchmarks

All six datasets are regularly sampled multivariate series drawn from four domains. We order them by channel effective rank, the axis along which the paper's scope condition is stated.

- **Synthetic** (D=8, er 7.99, $\overline{|\rho|}$ 0.007, 2000 windows). Multivariate sinusoids with per-channel frequency, phase and amplitude drawn independently, plus additive noise. The channels are constructed to be near-orthogonal, so it functions as the *reverse control* of this study: cross-channel information does not exist, and a method that gains from exploiting it should gain least here.
- **ETTm1** (D=7, 4.10, 0.224, 1451 windows) (Zhou et al., 2021). Electricity transformer load and oil temperature at 15-minute resolution. Seven physically distinct quantities with moderate coupling; the only dataset in the study whose channels are heterogeneous in units rather than being replicated sensors.
- **Traffic** (D=20, 2.84, 0.491, 365 windows) (Lai et al., 2018). Hourly road occupancy rates from freeway sensors in the San Francisco Bay Area. Of 862 sensors we retain the twenty most variable. Occupancy is bounded in $[0, 1]$ and strongly diurnal, so sensors on the same corridor co-vary but sensors on different corridors do not.
- **PEMS-BAY** (D=20, 1.55, 0.757, 1085 windows) (Li et al., 2018). Vehicle speeds at 5-minute resolution from 325 loop detectors in the same region, subsampled to twenty as above. Speed is far more spatially coupled than occupancy: a slowdown propagates along a corridor, which is why this dataset has an effective rank near that of Electricity despite a completely different domain.
- **Electricity** (D=20, 1.50, 0.791, 548 windows) (Dua & Graff, 2017). Hourly consumption of 321 clients, subsampled to twenty. Client loads share a strong common daily and weekly cycle, giving the second-lowest effective rank in the study.
- **Solar** (D=20, 1.17, 0.920, 543 windows) (Lai et al., 2018). Ten-minute production records from 137 photovoltaic plants in Alabama, subsampled to twenty. Plants in one state share the same solar cycle and largely the same weather, giving the lowest effective rank and the highest mean absolute channel correlation of any dataset here.

A.2 Preprocessing

Every dataset passes through an identical pipeline, and we apply it without dataset-specific tuning so that the differences between rows of Table 1 reflect the data rather than our choices. Channels

are ranked by raw variance and the top D retained; remaining NaNs are set to zero; each channel is standardised to zero mean and unit variance over the full series; the series is cut into non-overlapping windows of length T; windows are shuffled with seed 42 and capped at 3000. The split is $80/20$ and no validation split is held out, since early stopping is disabled for every method, so the entire 20% is the test set. Train/test window counts are $1600/400$ (Synthetic), $1160/291$ (ETTm1), $292/73$ (Traffic), $868/217$ (PEMS-BAY), $438/110$ (Electricity) and $434/109$ (Solar).

Normalisation statistics are taken over the full series, and this is a leak. The standardisation above precedes the split, so the test windows contribute to the mean and variance used to scale the training windows. Every method sees the identical pipeline, so the leak is symmetric and cannot move one arm relative to another, but it is a leak and we state it rather than leave it to be found. We reran a control on the two real datasets at the ends of our redundancy range, Electricity (e_r 1.50, where the per-variable prior inverts) and ETTm1 (4.10, where it does not, and which carries the sequence-length study), under the three settings that carry the account: Block 30%, Block 50% and MCAR 30%. The control recomputes the per-channel mean and variance on the training windows alone and applies them to both splits, changing nothing else. Table 4 gives all twelve cells. The published column is re-derived from the released artefacts rather than copied from Table 1, and reproduces it in every cell, so the control and the main grid rest on the same aggregation.

Table 4: Control on the normalisation protocol (MAE ↓). Each cell gives the published protocol and the same runs with the normalisation statistics computed on the training split alone. Lower is better; the best of the three arms in each row is in **bold** under both protocols.

		FM-INTERP		SAITS		GAPFLOW	
Dataset	Setting	published	train-only	published	train-only	published	train-only
Electricity	Block 30%	0.2259	0.2310	0.1660	0.1703	**0.1331**	**0.1354**
	Block 50%	0.2602	0.2668	0.1762	0.1801	**0.1418**	**0.1442**
	MCAR 30%	0.0935	0.0951	0.1223	0.1231	**0.0901**	**0.0917**
ETTm1	Block 30%	0.1066	0.1065	0.1272	0.1218	**0.0965**	**0.0960**
	Block 50%	0.1474	0.1469	0.1739	0.1726	**0.1262**	**0.1255**
	MCAR 30%	0.0760	0.0757	0.0817	0.0806	**0.0735**	**0.0732**

Nothing moves. GAPFLOW attains the lowest MAE in all six settings; the Block-against-MCAR contrast is -28.0% against -3.4%, where the published protocol gives -27.7% against -3.5%; and FM-INTERP still loses to the attention imputer on both Electricity Block rows, by 35.6% and 48.1% against the 37.2% and 47.7% of the published protocol. The leak is real and it changes nothing that this paper claims.

A.3 MISSINGNESS MECHANISMS

Four mechanisms are applied at two nominal ratios, giving eight settings per dataset.

- **MCAR.** Each entry is dropped independently with probability r. Expected run length of a gap is $1/(1-r)$, so at r=0.3 most gaps are one or two steps and runs beyond six are rare.
- **Block.** One contiguous run per channel of length $\lfloor rT \rfloor$, with a uniformly sampled start. At T=48 and r=0.3 this is a 14-step gap; at r=0.5 it is 24 steps, half the window.
- **MNAR.** Entries are dropped with probability increasing as the value decreases, so the observed set is biased upward. This is the mechanism under which the predictability gate is harmful (Appendix H).
- **Irregular.** Asynchronous per-channel sampling patterns, so different channels are observed at different steps.

Bit-identical evaluation masks. Each method generates its evaluation masks through its own data-loading path, seeded per sample. We compare SHA-256 digests of the resulting mask arrays across methods over all 48 dataset × mechanism × ratio combinations and confirm they agree. We separately confirm that the 48 digests are pairwise distinct, so the check cannot pass vacuously by

comparing a constant against itself. The same test asserts that the split expression is shared at source level.

A caveat on the Irregular rate. The Irregular generator adjusts an initial pattern by removing observations but never adding them, so its realised missing rate is 70–72% regardless of the nominal 30% or 50%. We leave the generator unchanged so that those rows remain comparable to the protocol as published, but the nominal rate should not be read as the realised one. All methods see identical masks, so the comparison within those rows is unaffected; only its label is.

B THE EXCLUDED SEVENTH DATASET

We ran a seventh dataset, multi-site Beijing meteorology (Chen, 2017) (D=20, er 2.15, $\overline{|\rho|}$ 0.545, 426 windows), and report it here rather than in the main grid. The reason concerns preprocessing, not results. The packaged source is already standardised per channel, so ranking 132 channels by variance compares values that are all approximately 1 and the ordering is decided by mantissa noise; we measured effective ranks of 5.49 and 6.48 for the same matrix in single and double precision, neither reproducible. Selecting by observation rate instead is stable but is a *different rule* from the one applied to the other six datasets, and it happens to select twenty meteorological channels with no pollutant among them. We therefore exclude it from the main grid on grounds of comparability, and report it in full here, including the settings where GAPFLOW does not win.

Table 5: The excluded seventh dataset, all eight settings (MAE ↓), with the same columns as Table 1. GAPFLOW is best in 7 of 8 against the nine published baselines, losing MNAR 50%, and in 3 of 8 once the per-variable prior of (2) is counted as well. This is the weakest dataset in the study and we report it in full. Best in **bold**, second best underlined.

		FLOW-MATCHING						DETERMINISTIC / GENERATIVE				
Mechanism	Rate	CLWF	FlowTS	FM-Interp	PSW-I	TG-MSFM	TSFlow	SAITS	BRITS	CSDI	GP-VAE	**Ours**
MCAR	30%	0.092	0.384	0.058	0.749	0.221	0.126	0.077	0.070	0.077	0.313	**0.057**
MCAR	50%	0.113	0.486	**0.074**	1.276	0.280	0.135	0.100	0.094	0.118	0.351	0.077
Block	30%	0.103	0.433	0.074	1.007	0.234	0.200	0.089	0.085	0.089	0.278	**0.065**
Block	50%	0.159	0.550	0.120	1.418	0.319	0.249	0.131	0.128	0.180	0.349	**0.105**
MNAR	30%	0.280	0.419	**0.131**	1.155	0.343	0.179	0.138	0.153	0.151	0.504	0.132
MNAR	50%	0.309	0.492	0.175	1.419	0.385	0.211	**0.174**	0.187	0.183	0.543	0.176
Irregular	30%	0.338	0.577	**0.108**	1.365	0.380	0.149	0.130	0.129	0.173	0.456	0.112
Irregular	50%	0.262	0.587	**0.112**	1.684	0.382	0.146	0.126	0.132	0.174	0.462	0.114

Table 5 gives the full grid. The single loss against a published method is MNAR 50%, to an attention imputer by 1.0%. The pattern matches the account of Section 3.2: an effective rank of 2.15 places this dataset in the middle of our range, and the selected meteorological channels are smoother and less mutually predictive than the replicated sensor arrays that drive the main result, so the margin over a per-variable prior is correspondingly small.

C IMPLEMENTATION DETAILS

Velocity network. A 4-layer pre-norm Transformer encoder with d_{model}=256, 4 heads, feed-forward width 1024, GELU activations, dropout 0.1 and learned positional embeddings, totalling ∼3.9M parameters. The input at each step is the current state x_t concatenated with the observation mask and a missingness-geometry channel; a sinusoidal embedding of the flow time is added to every position. Every flow-matching arm in this paper—GAPFLOW, FM-INTERP and the reimplemented baselines—uses this identical network, so capacity is not a confound within that family. The PyPOTS baselines retain their published architectures, which range from 0.01M to 0.55M parameters; we note this where the comparison matters.

Source priors. The interpolation prior of (2) uses σ_{obs}=0.1 and $\sigma_{\max}$=1.0. For (3), Σ is estimated once per run from the training data using every pair of channels wherever both are observed, with eigenvalues clipped below 10^{-6} and ridge λ=0.1. The gap-length weight of (4) uses τ=6 and the predictability gate uses θ=0.3. The path of (6) uses s_{near}=1.6, $s_{\min}$=1.0 and τ_s=6. None of these was tuned per dataset.

Training. AdamW with learning rate 10^{-3}, weight decay 10^{-4}, batch size 64, 500 epochs, five warm-up epochs followed by cosine decay, and gradient clipping at 1.0. Training missingness is drawn per sample from a random-rate MCAR schedule with $r \sim \mathcal{U}(0.1, 0.7)$, independent of the evaluation mechanism. The mask for sample i at epoch e is produced by a stateless generator seeded on (seed, e, i), so the training stream is reproducible and identical across arms rather than depending on data-loader ordering.

Inference. Euler integration with 20 steps uniform in t, matching the training target $u_t = \dot{\alpha}_t(x - z_0)$: the network regresses $\mathrm{d}x_t/\mathrm{d}t$, so the $\dot{\alpha}_t$ factor is carried by the target rather than by the step size. The reported MAE averages $K{=}20$ independent samples per window, following the protocol under which the baselines were evaluated. Self-conditioning runs $n{=}3$ rounds; a safeguard disables it when a window's observation rate falls below 0.3, which never occurs on any dataset reported here. All timings are measured on a single CPU node with 16 threads.

D Baselines, Fairness, and Evaluation Metrics

D.1 Baselines

We compare against nine external methods spanning deterministic, generative and flow-matching families. For each we state the reproduction route: `PyPOTS` denotes the PyPOTS library (Du, 2023), `Official` the authors' released code, and `Re-impl.` our faithful reimplementation following the published algorithm.

- **SAITS** (Du et al., 2023) combines a diagonally-masked self-attention block with a joint reconstruction and imputation objective, attending over time and features simultaneously to produce deterministic point estimates. Of the nine baselines it takes first place on 22 of our 48 main-grid rows, more than any other. Evaluated via `PyPOTS`.
- **BRITS** (Cao et al., 2018) runs forward and backward recurrent passes whose hidden states decay with the elapsed time since the last observation, and combines the two estimates through a consistency loss. Strongest baseline on 7 rows. Evaluated via `PyPOTS`.
- **CSDI** (Tashiro et al., 2021) applies conditional score-based diffusion with a Transformer backbone, denoising missing coordinates while clamping observed ones and producing calibrated uncertainty from repeated sampling. It requires 50 network evaluations per sample and carries a median 15% higher error than SAITS on our grid. Strongest baseline on 5 rows. Evaluated via `PyPOTS`.
- **GP-VAE** (Fortuin et al., 2020) is a variational autoencoder with a Gaussian-process prior over the latent trajectory, enforcing temporal smoothness and providing uncertainty over missing entries. On every dataset in our study it ranks in the bottom four of the nine, and last on Traffic; PSW-I is last on the other five. Evaluated via `PyPOTS`.
- **FlowTS** (Hu et al., 2024) adapts rectified flows to time series, learning a velocity field from Gaussian noise to the data distribution. The `Official` release covers generation only, so we `Re-impl.` imputation by conditioning on observed entries during ODE integration, following the paper's description.
- **CLWF** (Qian et al., 2024) learns a continuous-time transport map under a Lagrangian optimal-transport cost that encourages straight trajectories. With no official code available we build the imputer from the published algorithm (`Re-impl.`). Of all nine baselines this benefits most from our uniform 500-epoch budget, improving by 13.3% over its published 100-epoch setting.
- **TSFlow** (Kollovieh et al., 2025) places a Gaussian-process posterior as the source distribution of a conditional flow matcher. The `Official` release covers forecasting, so imputation is `Re-impl.` here. It leads the field on 14 of our rows and on none of the Block rows: all 14 are MCAR, MNAR or Irregular settings on Solar, Synthetic and PEMS-BAY, where gaps are short. A GP posterior is a per-variable construction in the same sense as the interpolation prior, and this is the pattern Section 3.2 predicts for one: competitive when a channel's own observations bracket the gap, and not when they do not.
- **PSW-I** (Wang et al., 2025) formulates imputation as an optimal-transport problem and minimises a Wasserstein objective over importance-weighted samples. Evaluated using the `Official`

codebase. Alone among the nine it *degrades* under our uniform budget, by 21.2% relative to its published setting.

- **TG-MSFM** (Wang & Sugiyama, 2026) gates a multi-scale velocity field on a normalised diffusion time and fuses temporal scales through a learned softmax. We `Re-impl.` it at our protocol's capacity rather than its published 64-dimensional configuration, which would understate it; we also evaluate it at $T{=}128$, its native window length, where it does not overtake GAPFLOW.

Fidelity of the reimplementations. To check that the reimplemented arms reproduce their published behaviour rather than a degraded variant, we run one method at its originally published epoch budget and compare against its published number on a shared setting: we obtain 0.12084 against 0.12083, a relative difference of 0.0035%. The harness sets no random seed anywhere on that code path, so the agreement reflects the code path rather than seed alignment.

D.2 FAIRNESS OF COMPARISON

Three asymmetries are worth stating, and two of them work against our reported margins.

- **Training missingness.** For MCAR and Block the reimplemented flow-matching baselines train on missingness matching the evaluation mechanism. GAPFLOW and FM-INTERP train on a mechanism-agnostic random-rate schedule and never see the test-time mechanism during training.
- **Checkpoint selection.** That harness selects its best checkpoint by loss on the *evaluation* split; ours selects on the training split.
- **Budget.** The uniform 500-epoch budget is not uniformly generous. Relative to the published 100 epochs, CLWF improves by 13.3%, FlowTS and TSFlow move by under 1.5%, and PSW-I degrades by 21.2%.

We retain all three rather than adjusting them, on the grounds that adjusting a protocol after seeing which direction the adjustment helps is the failure mode the rest of this appendix is designed to guard against.

Does the attention baseline have more to give? The deterministic baselines keep their published architectures, so one reading of the margins in Table 1 is that they reflect capacity rather than design. We tested that on the six settings the failure-mode claim rests on, retraining SAITS at 3 layers with $d_{\text{model}}{=}192$ and at 4 layers with 256, both for the same 500 epochs. The larger configurations help on some of the six, by under 5%, and on Solar they are worse than the published size. Taking the best configuration per setting, FM-INTERP still loses on all six and GAPFLOW still wins on all six. Making the attention imputer stronger widens the gap the paper reports rather than closing it, because the design that inverts is the structured prior, not the transformer. Table 1 keeps the published configuration so that every baseline is reported at the size its authors chose.

D.3 EVALUATION METRICS

Let $\mathcal{U} = \{(t,k) : m_{tk} = 0\}$ be the held-out set, x the ground truth, $\hat{x}$ the point estimate, and $\{S^{(1)}, \dots, S^{(K)}\}$ the K samples. All metrics are computed on $\mathcal{U}$ only.

Mean absolute error.

$$\text{MAE} \;=\; \frac{1}{|\mathcal{U}|} \sum_{(t,k)\in\mathcal{U}} \big|\hat{x}_{tk} - x_{tk}\big|, \qquad \hat{x} = \frac{1}{K}\sum_{k} S^{(k)}. \tag{8}$$

We compute it as the mean over evaluation batches of batch-level means. This coincides with the global mean when batches are equal-sized and differs by under 0.1% otherwise. The flow-matching arms and GAPFLOW share the convention, and each artefact records which one its run used.

Continuous ranked probability score.

$$\text{CRPS} \;=\; \frac{1}{|\mathcal{U}|} \sum_{(t,k)\in\mathcal{U}} \Big[\, \frac{1}{K}\sum_{i} \big|S^{(i)}_{tk} - x_{tk}\big| \;-\; \frac{1}{2K^2}\sum_{i,i'} \big|S^{(i)}_{tk} - S^{(i')}_{tk}\big| \,\Big]. \tag{9}$$

Average coverage error. For quantile levels $q \in \{0.05, 0.10, \ldots, 0.95\}$ with empirical quantiles Q_q of the samples,

$$\mathrm{ACE} \;=\; \frac{1}{|q|}\sum_{q}\Big|\;\frac{1}{|\mathcal{U}|}\sum_{(t,k)\in\mathcal{U}}\mathbf{1}\big[x_{tk}\le Q_{q,td}\big]\;-\;q\;\Big|. \tag{10}$$

Interval score at the 90% level. With $\ell = Q_{0.05}$, $u = Q_{0.95}$ and $\alpha = 0.1$,

$$\mathrm{IS}_{90} \;=\; \frac{1}{|\mathcal{U}|}\sum_{(t,k)\in\mathcal{U}}\Big[(u-\ell)+\tfrac{2}{\alpha}(\ell-x)_+ + \tfrac{2}{\alpha}(x-u)_+\Big]_{tk}. \tag{11}$$

Energy score. Treating each window as a vector $S^{(k)} \in \mathbb{R}^{TD}$,

$$\mathrm{ES} \;=\; \frac{1}{K}\sum_{k}\big\|S^{(k)}-x\big\|_2 \;-\; \frac{1}{2K^2}\sum_{k,k'}\big\|S^{(k)}-S^{(k')}\big\|_2. \tag{12}$$

CRPS and the energy score reward sharpness and accuracy jointly; ACE and IS_{90} penalise under-dispersion directly, which is why they are the two on which GAPFLOW is worse (Appendix G).

E ADDITIONAL METHOD DETAILS

Why two-sided distance rather than elapsed time. The elapsed-time feature δ increases monotonically through a gap and is largest at the step just before the gap closes, where the next observation is one step away and the interpolant is at its most reliable. The two-sided distance d of (5) peaks instead at the gap centre, which is where the chord is least reliable. Measured over the four twenty-channel datasets, δ overestimates d by 80% under Block missingness and by 27% under MCAR, with correlations of 0.22 and 0.40 respectively. The two features are least interchangeable in precisely the regime this paper is about.

Why pseudo-observations enter the prior only. Equation (7) substitutes previous-round estimates into the conditioning values of the cross-variable conditional but leaves the mask untouched. Modifying the mask would make the loss and the reported MAE partly self-referential, since positions the model had already filled would begin to count as evidence. Keeping the mask fixed means that $\mathcal{L}_{\mathrm{CFM}}$ and every reported number remain defined on the true missing set regardless of how many refinement rounds are run.

Cost of the covariance estimate. Estimating Σ is $O(ND^2)$ once per training run. The conditional solve in (3) is formed as one batched linear system over all $(\text{window}, \text{step})$ pairs, not grouped by observed-channel pattern. Grouping would buy little: on the twenty-channel datasets 52% of rows carry a distinct pattern under Block missingness and 97% under MCAR, and an implementation that groups by pattern and caches a Cholesky factor per group runs 20 to 100 times *slower* than the batched solve, because the per-group overhead dominates at these pattern counts. The batched solve costs 8–10 ms per batch of 64 windows at $D{=}20$ and 4 ms at $D{=}8$, which is 0.3–1.1 times a single velocity evaluation and under a tenth of one 20-step integration. Under our training missingness schedule, pairwise-complete estimation recovers between 38 and 46 times as many usable channel pairs as listwise deletion would, because a row of twenty channels is rarely complete when each is dropped independently at an average rate of 0.4.

F SEQUENCE LENGTH GENERALISATION

Table 3 in the main text reports ETTm1 at $T \in \{24, 96, 128\}$, the 24 settings omitted from Table 1. This section records the confound behind the shape of that result.

The gain grows with window length and then falls back: on Block 30% it runs -0.8%, -9.5%, -11.6% and -8.2% at $T{=}24, 48, 96, 128$. Two effects move in opposite directions here. A longer window means a longer contiguous gap at the same nominal ratio, which the account predicts should

favour us; it also means fewer windows (1451 at T=48 against 544 at T=128), which cuts the other way by reducing the data available to estimate Σ and to train the network. We registered this confound before running T=128 and cannot separate the two effects with the data we have; disentangling them would require holding window count fixed while varying T, which no fixed-length series permits without overlapping windows.

G Uncertainty Calibration: A Negative Result

GapFlow improves point accuracy and does not improve calibration. Over the 48 settings at T=48 it attains the better value in 27/48 for CRPS, 24/48 for the energy score, 12/48 for IS_{90} and 8/48 for ACE. The first two of those read as a coin flip and are not one: the aggregate averages over mechanisms whose behaviour is opposite. Table 6 separates them and Table 7 gives every setting.

All four metrics are computed from the same runs as Table 1—the same arm, the same checkpoints, the same three seeds and the same K=20 samples per window—using the definitions of Appendix D. The MAE of these runs reproduces the main table, 46 of 48 with the same two MCAR exceptions, so the point and the probabilistic numbers describe one model rather than two.

Table 6: Probabilistic metrics over the 48 settings at T=48, GapFlow against FM-Interp; all four are lower-is-better. "Wins" counts the settings in which GapFlow attains the lower value and "med. Δ" is the median of the per-setting relative changes (blue improvement, orange regression). CRPS and the energy score reward sharpness and accuracy jointly; IS_{90} and ACE penalise under-dispersion directly, and the sign of the effect flips between the two pairs. For reference, on the same runs GapFlow attains the lower MAE in 46/48, median −4.8%. The median of the per-setting changes is not the change between the medians: the energy score's median value falls from 2.50 to 2.34 while its median per-setting change is −0.1%, because the score's scale varies by an order of magnitude across datasets.

	CRPS		Energy score		IS_{90}		ACE	
Mechanism	Wins	med. Δ	Wins	med. Δ	Wins	med. Δ	Wins	med. Δ
Block	10/12	−11.9%	10/12	−9.1%	5/12	+7.0%	0/12	+88.1%
Irregular	10/12	−1.4%	9/12	−2.2%	7/12	−1.7%	4/12	+17.3%
MNAR	6/12	−0.4%	5/12	+1.4%	0/12	+28.8%	2/12	+54.6%
MCAR	1/12	+2.9%	0/12	+5.7%	0/12	+35.6%	2/12	+292.9%
All 48	27/48	−0.9%	24/48	−0.1%	12/48	+19.2%	8/48	+77.6%

The two sharpness-rewarding metrics follow the point accuracy. Under Block missingness GapFlow attains the lower CRPS in 10 of 12 settings, median −11.9%, and the lower energy score in 10 of 12, median −9.1%; on Electricity at 50% the two reach −42.9% and −38.0%. Under MCAR, where the point-accuracy gain is already ≈0%, the same refinement is pure cost: CRPS is better in 1 of 12 and worse by 2.9% at the median, the energy score in 0 of 12 and worse by 5.7%. The mechanism-dependence that organises the rest of the paper therefore organises the probabilistic metrics as well, and MCAR is the worst mechanism on all four of them. The correspondence is not an exact one and we do not claim it: under MAE the ordering is Block, then MNAR, then Irregular, then MCAR, whereas under CRPS and the energy score the middle two exchange places, and under IS_{90} and ACE it is Irregular rather than Block that fares best. We have no account of the exchange.

ACE and IS_{90} run the other way, and no mechanism rescues them. ACE is worse under every mechanism—0 of 12 under Block at a median of +88.1%, +292.9% under MCAR—and worse in 40 of the 48 settings. IS_{90} is worse in 36 of 48 and improves only under Irregular sampling (7/12, −1.7%). The four settings where the point-accuracy gain is largest are the sharpest statement of the trade: on Electricity and Traffic under Block missingness, CRPS improves by 23.7 to 42.9%, the energy score by 12.1 to 38.0% and IS_{90} by 7.7 to 38.6%, while ACE degrades by 10.7 to 30.0% on all four. There is no regime in this study in which GapFlow is better calibrated than FM-Interp.

The mechanism is the refinement itself. Each self-conditioning round substitutes the previous round's estimate into the conditioning values, so the K samples are drawn from priors that have

already been pulled towards a common point. Point accuracy improves and sample spread narrows; the predictive distribution becomes under-dispersed, which ACE and IS_{90} penalise directly and which CRPS and the energy score penalise only weakly because both also reward sharpness. The pattern across the four metrics is therefore not four independent failures but one failure seen through four lenses of differing sensitivity—and, read from the other end, one gain seen through the two lenses that reward sharpness.

What is new here, and what is entailed. CRPS is dominated by its first term, $\frac{1}{K}\sum_i |S^{(i)} - x|$, so a large CRPS improvement under Block missingness restates the MAE improvement more than it corroborates it, and we do not count that column as independent evidence. Two of the four results are not restatements. The energy score is taken on the whole window as a vector rather than coordinate by coordinate, so its -9.1% against $+5.7\%$ contrast says the joint structure of the imputed window improves under contiguous gaps and not only its marginals. And the reversal of IS_{90} and ACE is information no point metric can carry at all.

Scope. Every comparison in this section is against FM-INTERP, our own starting point rather than a published method. We did not evaluate the nine published baselines on probabilistic metrics, and nothing here supports a claim about GAPFLOW's calibration relative to them.

What to do about it. A practitioner who needs calibrated intervals rather than accurate points should set $n{=}1$, which costs one third of the network evaluations and recovers the spread of the underlying flow. We did not attempt a variance correction. The obvious candidates are inflating the prior variance in later rounds, or ensembling across rounds rather than replacing; both would move the point estimate as well, so the trade-off is unlikely to be free.

H ABLATION DETAILS

Component contributions across all four mechanisms. Table 2 reports Block and MCAR. Across all four mechanisms the median relative changes are: cross-variable prior -13.0% (Block), -0.1% (MCAR), $+2.5\%$ (MNAR), 0.0% (Irregular); gap-adaptive path -2.4%, -1.2%, -2.7%, -2.0%; self-conditioning -6.8%, $+0.3\%$, -7.6%, $+3.9\%$; R^2 hinge $+0.1\%$, -0.1%, $+0.2\%$, -3.5%. Only the gap-adaptive path helps under every mechanism.

When does self-conditioning stop helping? Training once and then evaluating the same judgment path at $n \in \{1, 2, 3\}$ separates the refinement from everything else. Over the six datasets the median change from $n{=}1$ to $n{=}3$ is -6.5% under Block and -5.2% under MNAR, against -0.05% under MCAR and -0.05% under Irregular. The mechanism-dependence that organises the rest of the paper reappears here: the rounds pay for themselves exactly where a gap is long enough that the first pass has little to condition on. Counting which round is best per setting, $n{=}3$ wins four of six Block settings and five of six MNAR settings. Under MCAR and Irregular the three choices split almost evenly, and the occasional regression at $n{=}3$ stays under 2%. A practitioner who knows the mechanism can stop at $n{=}2$ and keep most of the gain at two thirds of the cost; we report $n{=}3$ throughout because fixing one value across all settings is the more conservative protocol.

The R^2 hinge is mechanism-selective. Measured as the final step of the ladder of Table 2, configuration (E) against configuration (D), over the twelve settings of each mechanism, the hinge helps in $11/12$ MCAR settings (median -0.1%) and $12/12$ Irregular settings (median -3.5%), and is mildly harmful on Block ($3/12$, median $+0.1\%$) and on MNAR ($4/12$, median $+0.2\%$). Both penalties are an order of magnitude smaller than the Irregular gain. The MNAR sign nonetheless has a candidate explanation: MNAR removes low values, so the observed set is biased upward, the conditional of (3) inherits that bias, and the hinge increases reliance on a biased predictor exactly where the bias is largest. This is a hypothesis consistent with the sign and with the mechanism selectivity, but not with any large effect and not tested directly; a direct test would compare hinge behaviour under a value-dependent mechanism with the bias removed by reweighting.

Why the ladder is ordered as it is. Incremental ablations are order-dependent, and a different ordering would attribute the shared variance differently. We fix the order by dependency: the

Table 7: Per-setting probabilistic metrics at $T=48$, FM-INTERP against GAPFLOW in each pair, with the row order of Table 1; "FM-I" is FM-INTERP and "Ours" is GAPFLOW. All four metrics are lower-is-better and the better of each pair is in **bold**. Values are rounded to three decimals and three settings tie at that precision without tying at full precision, so win counts recomputed from this table can differ by one from Table 6.

			CRPS		Energy score		IS_{90}		ACE	
Dataset	Mechanism	Rate	FM-I	Ours	FM-I	Ours	FM-I	Ours	FM-I	Ours
Synthetic	MCAR	30%	**0.072**	0.076	**0.977**	1.027	**0.576**	0.812	**0.024**	0.097
	MCAR	50%	**0.080**	0.085	**1.428**	1.529	**0.627**	0.912	**0.017**	0.095
	Block	30%	**0.261**	0.266	**3.906**	4.131	**2.266**	2.695	**0.055**	0.088
	Block	50%	**0.332**	0.350	**6.278**	6.793	**2.818**	3.199	**0.058**	0.067
	MNAR	30%	**0.081**	0.083	**1.093**	1.110	**0.677**	0.931	**0.046**	0.111
	MNAR	50%	0.093	**0.090**	1.631	**1.592**	**0.769**	1.007	**0.047**	0.107
	Irregular	30%	0.148	**0.147**	3.705	**3.568**	**1.209**	1.254	**0.018**	0.043
	Irregular	50%	0.145	**0.144**	3.671	**3.617**	1.184	**1.159**	**0.018**	0.019
ETTm1	MCAR	30%	0.056	0.056	**0.824**	0.829	**0.473**	0.590	**0.013**	0.047
	MCAR	50%	**0.063**	0.064	**1.217**	1.247	**0.540**	0.702	**0.012**	0.058
	Block	30%	0.079	**0.075**	1.125	**1.066**	**0.709**	0.846	**0.030**	0.073
	Block	50%	0.111	**0.101**	2.082	**1.889**	**1.119**	1.256	**0.044**	0.095
	MNAR	30%	0.086	**0.085**	1.325	**1.312**	**0.817**	1.031	**0.039**	0.080
	MNAR	50%	0.121	**0.115**	2.614	**2.486**	**1.308**	1.552	**0.056**	0.094
	Irregular	30%	0.075	**0.074**	**1.699**	1.702	**0.641**	0.718	**0.012**	0.036
	Irregular	50%	0.075	**0.074**	1.734	**1.714**	0.644	**0.636**	**0.012**	0.014
Traffic	MCAR	30%	**0.082**	0.092	**3.007**	3.615	**0.767**	1.173	**0.015**	0.062
	MCAR	50%	**0.099**	0.109	**4.727**	5.591	**0.941**	1.446	**0.016**	0.081
	Block	30%	0.186	**0.142**	6.867	**5.818**	2.170	**1.954**	**0.062**	0.080
	Block	50%	0.224	**0.171**	9.813	**8.628**	2.642	**2.440**	**0.079**	0.102
	MNAR	30%	**0.175**	0.176	**6.113**	6.391	**1.744**	2.409	**0.061**	0.079
	MNAR	50%	0.337	**0.298**	14.261	**13.471**	**3.889**	4.288	**0.111**	0.131
	Irregular	30%	**0.122**	0.125	**6.668**	7.027	**1.185**	1.451	**0.027**	0.074
	Irregular	50%	0.122	**0.117**	6.875	**6.636**	1.179	**1.111**	**0.026**	0.033
PEMS-BAY	MCAR	30%	**0.043**	0.044	**1.170**	1.235	**0.377**	0.481	**0.018**	0.064
	MCAR	50%	**0.048**	0.050	**1.733**	1.847	**0.420**	0.560	**0.016**	0.074
	Block	30%	0.094	**0.086**	2.857	**2.705**	**1.028**	1.110	**0.043**	0.102
	Block	50%	0.116	**0.099**	4.540	**4.132**	1.365	**1.345**	**0.051**	0.113
	MNAR	30%	**0.064**	0.066	**1.919**	1.995	**0.651**	0.834	**0.123**	0.152
	MNAR	50%	**0.084**	0.088	**3.684**	3.911	**0.965**	1.234	**0.097**	0.135
	Irregular	30%	0.057	0.057	2.498	**2.489**	**0.509**	0.512	**0.019**	0.023
	Irregular	50%	0.057	0.057	**2.516**	2.529	**0.506**	0.513	**0.018**	0.023
Electricity	MCAR	30%	**0.069**	0.071	**1.776**	1.881	**0.620**	0.855	**0.023**	0.088
	MCAR	50%	**0.076**	0.077	**2.505**	2.628	**0.688**	0.952	**0.025**	0.095
	Block	30%	0.177	**0.109**	4.246	**2.809**	2.206	**1.482**	**0.109**	0.126
	Block	50%	0.205	**0.117**	6.250	**3.876**	2.642	**1.623**	**0.121**	0.134
	MNAR	30%	0.081	**0.080**	**2.164**	2.192	**0.788**	1.020	**0.050**	0.101
	MNAR	50%	0.103	**0.096**	3.544	**3.385**	**1.103**	1.295	**0.063**	0.118
	Irregular	30%	0.089	**0.084**	3.434	**3.252**	0.841	**0.774**	0.044	**0.038**
	Irregular	50%	0.090	**0.084**	3.508	**3.306**	0.853	**0.778**	0.045	**0.038**
Solar	MCAR	30%	0.031	**0.030**	**0.855**	0.880	**0.277**	0.340	0.075	**0.042**
	MCAR	50%	**0.036**	0.037	**1.313**	1.396	**0.333**	0.444	0.074	**0.068**
	Block	30%	0.069	**0.063**	1.743	**1.597**	**0.718**	0.859	**0.030**	0.066
	Block	50%	0.081	**0.066**	2.573	**2.150**	**0.863**	0.916	**0.034**	0.091
	MNAR	30%	**0.040**	0.041	**1.112**	1.139	**0.391**	0.507	0.072	**0.035**
	MNAR	50%	**0.051**	0.052	**1.795**	1.848	**0.524**	0.703	0.067	**0.042**
	Irregular	30%	0.044	**0.042**	1.820	**1.767**	0.396	**0.380**	0.057	**0.040**
	Irregular	50%	0.044	**0.042**	1.872	**1.819**	0.403	**0.386**	0.058	**0.041**

gap-adaptive path uses the two-sided distance that the cross-variable blend also needs, and self-conditioning requires a cross-variable prior to feed pseudo-observations into. The R^2 hinge is placed last because it modifies the blend that all earlier components have already established. We did not run the order-independent alternative, a leave-one-out ablation, so the magnitudes we attribute to each component are conditional on this ordering; the ranking of the three load-bearing components should be read as an ordering under one dependency-respecting sequence rather than an order-free measurement.

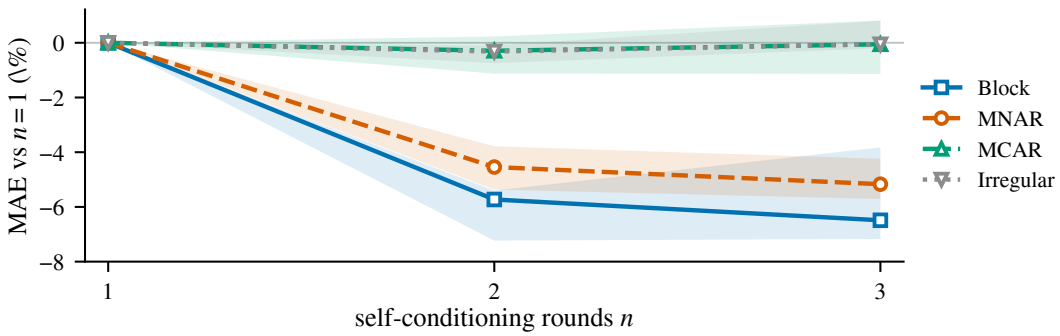

Figure 3: Effect of the refinement rounds, by mechanism: median over the six datasets, shaded to the interquartile range. The model is trained once and the same judgment path is evaluated at each n, so the differences carry no retraining noise. The rounds buy 5–7% under Block and MNAR and nothing under MCAR and Irregular, and most of what they buy arrives by $n{=}2$.

I Accuracy–Cost Trade-off

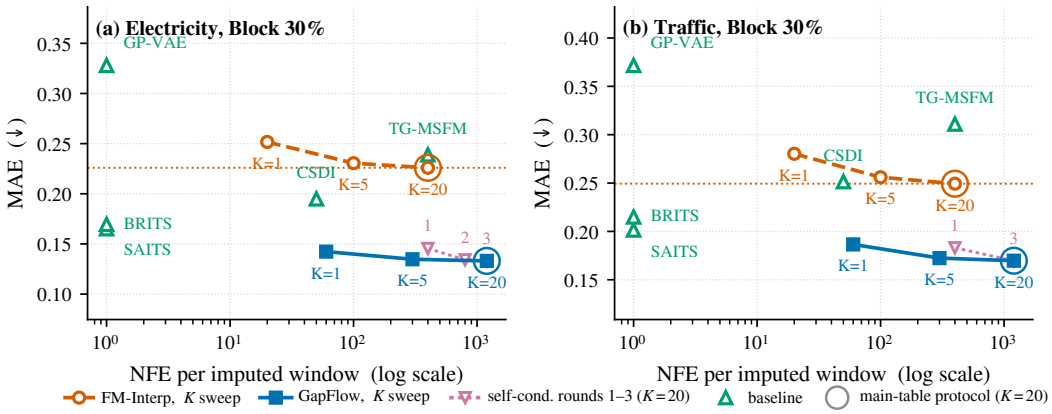

Figure 4: Accuracy against network evaluations per window. Circled markers are the main-table protocol. At one sample GapFlow costs 60 evaluations and still beats FM-Interp at twenty.

Network evaluations per imputed window, measured by a counter attached to the forward pass rather than read off the code: 1 for the deterministic baselines, 50 for the diffusion imputer, 400 for FM-Interp and TG-MSFM, and 1200 for GapFlow. The 3.0× factor over FM-Interp decomposes exactly as 20 integration steps × $K{=}20$ samples × $n{=}3$ refinement rounds, and we verify the decomposition by perturbing each factor and confirming the counter tracks it.

Table 8: Inference cost per imputed window, Block 30%. Network evaluations (NFE) are hardware-independent and identical across repeats; wall-clock is the median over 2 timing runs on 16 CPU threads and carries up to 12% spread between runs, so ratios are more reliable than absolute values. Parameter counts are the models as run.

		Electricity		ETTm1	
Method	Params	NFE	ms/window	NFE	ms/window
SAITS	0.55M	1	0.3	1	0.2
BRITS	0.19M	1	5.2	1	5.0
GP-VAE	18K	1	0.4	1	0.4
CSDI	0.30M	50	206	50	51
TG-MSFM	3.87M	400	249	400	209
FM-Interp	3.90M	400	227	400	188
GapFlow ($n{=}1$)	3.90M	400	201	400	188
GapFlow ($n{=}2$)	3.90M	800	410	800	382
GapFlow ($n{=}3$, ours)	3.90M	1200	627	1200	574

Table 8 puts network evaluations, wall-clock and parameter count side by side, since none of the three alone describes what a deployment pays. Two things are worth reading off it. The $3.0\times$ evaluation ratio over FM-INTERP translates to $2.8\times$ and $3.1\times$ in wall-clock on the two datasets, so the counter is not hiding per-call overhead. And at one refinement round GAPFLOW runs at the same evaluation count as FM-INTERP and slightly faster in wall-clock (201 against 227 ms on Electricity), because the cross-variable prior starts closer to the data and the integrator spends less time on large steps. The deterministic baselines are one to two orders of magnitude cheaper and one order smaller; the diffusion imputer sits between the two families at 50 evaluations.

Figure 4 plots the trade-off. The dominant term is K, a protocol inherited from the evaluation setup rather than a requirement of the method. Reducing K from 20 to 1 costs GAPFLOW a median 12.1% in MAE but costs FM-INTERP 31.3%, a $2.6\times$ larger penalty; at $K{=}5$ the two figures are 2.1% and 6.4%. The refinement rounds already perform part of the variance reduction that sample averaging otherwise provides. They do so where it matters: the $K{=}1$ penalty is 8.3% for GAPFLOW against 12.5% for FM-INTERP under Block, and 13.0% against 36.3% under MCAR, where the refinement has least to add. At $K{=}1$, GAPFLOW needs 60 evaluations, the same order as the diffusion baseline, and on Block settings still attains lower MAE than FM-INTERP at $K{=}20$, at one sixth of its cost. Practitioners with a fixed evaluation budget should therefore reduce K before reducing n.

J EXTENDED ANALYSIS

J.1 TRANSPORT GEOMETRY

Accumulating $\|\mathrm{d}z\|$ along the integration path and separating observed from missing coordinates, the ratio of missing-coordinate to observed-coordinate arc length has median 18.2 for FM-INTERP and 16.4 for GAPFLOW. Both structured priors concentrate transport on the missing entries by more than an order of magnitude, which is the mechanism by which they beat a Gaussian source; the slight reduction under GAPFLOW is consistent with its prior starting closer to the target inside long gaps, leaving less distance to travel. The ratio is measured on a single integration pass, so it characterises the base transport and does not include the refinement rounds.

J.2 QUALITATIVE FIGURE: HOW THE WINDOW WAS CHOSEN

Figure 2 shows one window per dataset, and a single window can be made to say almost anything, so the selection rule is stated in full. A candidate is a (window, channel) pair with a contiguous gap of at least ten steps whose two endpoints are both observed, since a chord is undefined otherwise. Among candidates we take the one whose ground truth has the largest mean absolute deviation from the straight chord through those endpoints, ties broken by window index. The rule reads only the ground truth and the mask; no method's prediction enters it.

That rule is not neutral: it selects where a per-variable prior is furthest from correct, which is the regime this paper is about but also the regime that favours GAPFLOW. The figure therefore answers "what does the failure look like", and the population statistics answer "how often". Electricity has 1386 qualifying gaps and Traffic 914; the median reduction of GAPFLOW over FM-INTERP inside them is 40.3% and 28.0%, and GAPFLOW is ahead in 84.3% and 77.1% of them. Two earlier rules we tried and discarded are recorded in the figure script: taking the longest gap selects a window with four surviving observations, and taking the median gap length selects a span over which the series is nearly flat and all three methods coincide.

The predictions in the figure come from the same evaluation function that produces Table 1. As a check, the MAE recomputed from the dumped arrays agrees with the main-table entry to 0.31% and 0.12% for FM-INTERP and GAPFLOW on Electricity and to 0.43% and 0.01% on Traffic, the residual being cross-node floating-point reduction order. The attention baseline is retrained rather than reloaded, since its harness does not fix a seed; it lands within 0.53% of its tabulated value.

J.3 GAP-BUCKETED ANALYSIS

Figure 5 bins imputed positions by two-sided distance d. Under Block missingness the relative gap between FM-INTERP and GAPFLOW widens monotonically: 9.5% at $d \in [1, 2]$, 18.7% at $[3, 6]$,

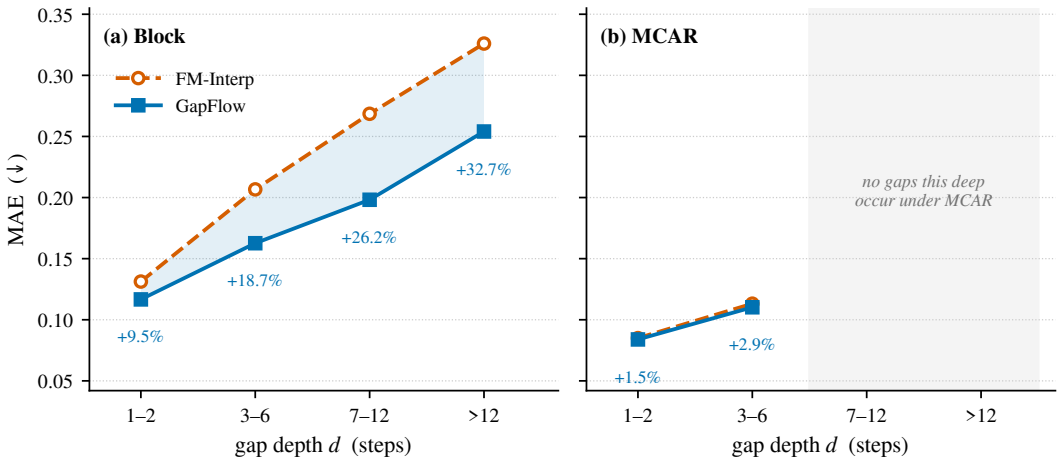

Figure 5: MAE against gap depth d, the two-sided distance to the nearest observation, pooled over the six datasets and both ratios. **(a)** Under block missingness the two methods are nearly identical at $d \leq 2$ and separate monotonically as the gap deepens, reaching a 32.7% relative gap beyond 12 steps: the per-variable interpolant degrades with gap length while cross-variable conditioning does not. **(b)** Under MCAR the curves overlap and the deep buckets are empty, because MCAR does not produce gaps that deep. Buckets are pooled *within* a mechanism only; pooling across mechanisms would confound gap depth with mechanism.

26.2% at $[7, 12]$ and 32.7% beyond 12. Under MCAR the corresponding values are 1.5% and 2.9% for the two buckets that contain enough positions to report. The figure uses all 144 runs of the two arms across the six datasets.

A bucket enters the figure only if all 12 dataset $\times$ ratio cells have data and every individual run contributes at least 200 positions to it; the threshold was fixed before inspecting the values. Under MCAR the two deepest buckets fail the first condition outright, because independent dropout at 30–50% almost never produces a run of seven or more consecutive missing steps within a single channel. We leave them empty rather than pooling across mechanisms. Pooling would confound gap depth with mechanism: short buckets are populated almost exclusively by MCAR and MNAR and deep ones almost exclusively by Block, so a pooled curve would measure the mechanism mix as much as the depth.

K Multi-Seed Evaluation

GapFlow and FM-Interp are run with three seeds (42, 123, 456) in every setting. Across the 48 cells of Table 1 the coefficient of variation of GapFlow has median 0.67% and maximum 7.01%. The largest value occurs on Synthetic Block, the setting with the largest absolute MAE and the fewest usable channel correlations, where the cross-variable term contributes least and the run-to-run variation of the velocity network dominates. Over the same 48 cells the margin over the strongest baseline has median 17.9% and minimum 4.2%; these are not the 16.4% and 3.2% of Section 5.1, which are taken over all 72 settings and therefore include the sequence-length grid. Comparing the two summaries to each other would say little, because the spread and the margin vary together across cells. The comparison that carries the point is per cell: in all 48 the margin exceeds that cell's own coefficient of variation, by a median factor of 23.6 and by at least 2.0. The tightest ratio is Synthetic Block 50%, where a margin of 11.1% sits against a spread of 5.5%; the noisiest cell, Synthetic Block 30% at 7.01%, still clears its own spread by a factor of 2.4.

The main grid uses the same three seeds in every cell. Two settings, Electricity Block 30% and 50%, were later rerun with two further seeds; averaging over five instead of three moves FM-Interp by -1.4% and -1.3% and leaves the coefficient of variation at 2.9% either way, so the extra runs change no comparison in the paper. We keep the tables at three seeds so that every cell rests on the same number of runs, and record the two five-seed settings here rather than mixing sample sizes across a grid whose cells are read against each other.

Every entry of Tables 1, 3 and 5 is the mean over the runs available for that cell, with no spread printed: the number of runs is not constant across columns, so a $\pm$ would have meant different things in different cells. TG-MSFM carries three seeds in all settings and the PyPOTS baselines carry between one and five depending on the setting, so we state the spread here, for the two arms where it is measured on a common footing, rather than attaching it unevenly to a grid.

L PRE-REGISTRATION

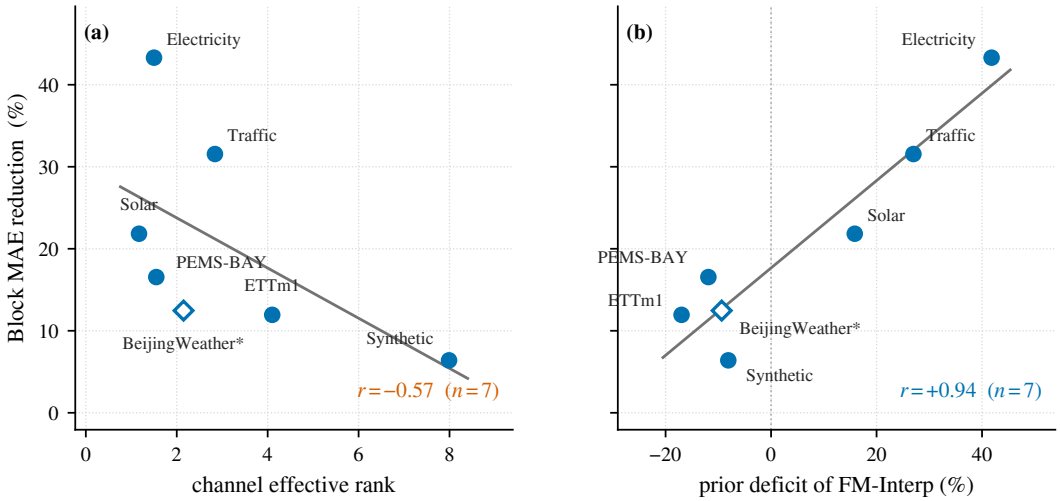

Figure 6: **(a)** Block-mechanism gain against channel effective rank: the direction is right but the relationship is weak, and it weakened as validation datasets were added, which is why P1 is recorded as falsified. **(b)** The same gains against how far FM-INTERP falls behind the strongest baseline, a relationship found after the fact and reported as such. Diamonds mark the excluded seventh dataset.

Four of the six datasets were added after the first version of this study. Before running them we recorded ten predictions with their decision rules, the data checksums, and the channel statistics on which the predictions rested. Six held and four were falsified. The falsified ones are:

- **P1** predicted that FM-INTERP would lose to an attention imputer on PEMS-BAY Block, as it does on Electricity. It won, by 16.8% and 7.0%.
- **P2** predicted a GAPFLOW gain of at least 30% on PEMS-BAY Block. We measured 14.1% and 19.0%.
- **P5** predicted a gain within $[31\%, 41\%]$ on the seventh dataset. We measured 12.7% and 12.2%.
- **P6** predicted a gain of at least 41% on Solar, our only extrapolation beyond the observed range of effective rank. We measured 17.0% and 26.7%.

All four rest on the same withdrawn hypothesis: that channel effective rank predicts the size of the gain. The surviving predictions are the mechanism ones, chiefly that the MCAR gain would fall below half the Block gain on every dataset, which held six times out of six and which we had registered as the falsification lever for the entire account. Had gains been uniform across mechanisms, the account would have been wrong regardless of the headline numbers.

One further relationship, shown in Figure 6b, was found in the data after the first falsification rather than before it. We registered it as an out-of-sample test before running the remaining datasets, and it held on all four held-out settings. We report it as an observation rather than a claim, because a relationship that was found in data and then tested on four additional points is weaker evidence than one that was stated in advance.

M NEGATIVE AND NEUTRAL DESIGN CHOICES

- **d-dependent prior mean and variance.** Scheduling the interpolation mean and the prior variance by d, rather than only the path speed, degrades end-to-end MAE by 23.5% and 19.1% despite

improving the prior in isolation by 6.9% at long gaps. The prior becomes organised by d while the network still receives δ, and the two correlate at only 0.22 under Block, so the regression target acquires variance the network cannot explain.

- **Replacing δ with d as a network input.** Substituting the geometry channel costs 5.1%. Providing both is better than either at the cost of extra parameters, so we report the substitution as neutral-to-negative and do not adopt it.
- **Observation-rate safeguard.** Never fires on any reported dataset, since observation rates run 50–70% against a threshold of 0.3. Retained for sparser regimes; contributes exactly 0.00% to every number here.
- **Ledoit–Wolf shrinkage** for Σ in place of ridge regularisation: no consistent improvement, and slower.
- **Windowed cross-variable conditioning**, restricting (3) to a temporal neighbourhood rather than the same step: neutral.
- **Calibration.** See Appendix G: point accuracy improves, calibration does not.

N Reproducibility

Every number in the paper is derivable from the released result artefacts and table generators without access to model weights or training code. The artefacts record per-run MAE together with the dataset, mechanism, ratio, arm, seed and epoch budget, and the generators reproduce every table from them; we verify this by unpacking the released archive into a clean directory and re-running the generators there. Evaluation masks are drawn by a generator seeded on the window index; their SHA-256 digests are recorded for all 48 settings and asserted in the released tests, which further confirm that the two harnesses agree cell by cell and that the 48 digests are pairwise distinct. Regenerating the masks requires the dataset loaders, which are not part of the archive, so that assertion is reported as skipped there. The probabilistic metrics and the gap-bucketed analysis are recomputed from stored per-sample outputs.

Two properties of the protocol are worth stating for anyone reproducing it. First, the training missingness schedule is independent of the evaluation mechanism, so one (dataset, arm, seed) triple trains a single model that is then evaluated under eight mask configurations; a reproduction that retrains per mechanism will spend eight times the compute for identical results. Second, the window-length variants are distinguished only by a filename suffix and not by a field inside the artefacts, so any aggregation must parse the suffix or the four lengths will silently collapse into one.