LEARNING FROM GENERATED CONTEXTS: ADAPTIVE STAGEWISE TRAINING FOR LONG-HORIZON FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-horizon forecasts of physical and dynamical systems accumulate errors as generated inputs evolve during recursive deployment. We study stagewise training: successive predictors learn true futures from contexts generated by frozen predecessors. We analyze specialization gains and error propagation; population bounds motivate monitoring without prescribing individual error curves. We introduce public-boundary adaptation and sample-specific adaptation, distinguishing model training from inference-time routing. Public adaptation uses aggregate error signals to allocate a compact model chain. Sample-specific training aligns successor windows with individual slope-based handovers; inference policies are then compared on the same frozen chain. On the tokamak benchmark, public slope control uses 30–40% fewer models than a ten-model baseline, with mean-error changes from -3.35% to $+2.82\%$. On engine and chemical benchmarks, model savings can accompany substantial accuracy losses. A diagnostic that switches when the adjacent successor reduces true scaled root mean squared error (sRMSE) by at least five percent improves mean error and $q90$/$q95$ at forecast lead 500 over learned routing on this benchmark, but requires future targets and is not uniformly best across all baselines. Learned error predictors fail to recover these gains. Across three datasets and three backbones, generated-context training reduces step-500 mean sRMSE by 17.63%–96.24% in 27 matched comparisons. Reliable individual routing remains unresolved.

1 INTRODUCTION

Long-horizon prediction matters in physical and dynamical systems because decisions often depend on how a state evolves before new measurements arrive. Learned weather forecasts and physical simulators support planning and repeated evolution calculations (Lam et al., 2023; Sanchez-Gonzalez et al., 2020), while chaotic systems expose the difficulty of extending local accuracy over long intervals (Pathak et al., 2018). Local accuracy can deteriorate when predictions repeatedly become inputs.

We study forecasting from a finite observed history without assimilating new observations. A predictor produces an output block and feeds it into the next input. Local errors then propagate, and the resulting generated contexts depart from the training distribution. Existing simulators and neural PDE solvers address aspects of this accumulation through training and refinement (Sanchez-Gonzalez et al., 2020; Lippe et al., 2023). We ask a complementary question: *how long should the same parameters be reused before an evolving context justifies a separately adapted successor?*

Our *adaptive stagewise training* framework (Figure 1) varies shared predictors' reuse durations. Public adaptation uses aggregate validation errors; sample-specific adaptation aligns training with individual handovers. On MAST, comparing inference policies on one frozen, slope-trained chain isolates routing from fitting and reveals a substantial gap between true-error comparison and learned routing.

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.

Our contributions are threefold:

- **Stagewise training and theoretical analysis.** We formulate learning on generated contexts and characterize specialization gains, error propagation, and conditional routing reliability, distinguishing population bounds from individual error curves.
- **Two adaptive allocation methods.** We develop public and sample-specific model reuse, including handover-aligned training, and show that public adaptation can approach fixed ten-model accuracy with fewer models, while exposing controller-dependent failures.
- **A diagnosis of individual routing.** On MAST frozen sample-specific chains, we compare learned routing with future-informed adjacent-model selection. The latter improves endpoint mean and tail errors over learned routing, identifying a gap between successor potential and deployable selection.

2 Related Work

Physical and dynamical forecasting. Reservoir computing predicts spatiotemporal chaos (Pathak et al., 2018), Fourier neural operators learn PDE solution mappings (Li et al., 2021), and GraphCast forecasts global weather (Lam et al., 2023). Graph simulators train on perturbed inputs (Sanchez-Gonzalez et al., 2020); PDE-Refiner addresses neglected frequencies and estimates rollout reliability (Lippe et al., 2023). We focus on the training and reuse duration of successive parameter sets. For our tokamak application, TokaMark (Rousseau et al., 2026) curates reconstruction and forecasting tasks from the FAIR-MAST diagnostic resource (Jackson et al., 2024); we use its data without proposing a plasma-specific architecture.

Architectures and pretraining. Forecasting architectures develop probabilistic and structured outputs (Salinas et al., 2020; Oreshkin et al., 2020; Lim et al., 2021), efficient attention and decomposition (Zhou et al., 2021; Wu et al., 2021; Zhou et al., 2022), and temporal or variate tokenization (Nie et al., 2023; Liu et al., 2024a); linear baselines emphasize evaluation design (Zeng et al., 2023). Pretrained models extend transfer across forecasting tasks (Das et al., 2024; Ansari et al., 2024; Woo et al., 2024; Liu et al., 2024b; 2025), while Time-MoE scales sparse experts (Shi et al., 2025). These advances do not themselves specify handovers between separately adapted predictors. Feedback error concerns recursive deployment, not every direct forecaster.

Learning on generated states. Scheduled Sampling mixes generated inputs into training, Professor Forcing aligns teacher-forced and free-running dynamics, and DAgger collects supervision on learner-induced states (Bengio et al., 2015; Lamb et al., 2016; Ross et al., 2011). DAgger's interactive guarantees do not directly transfer to offline physical forecasting. Flipped Classroom studies training curricula (Teutsch & Mäder, 2022); AROpt penalizes departures from an error-growth heuristic (Li et al., 2026). We additionally adapt which generated-context regimes receive separate parameters and how long each is reused.

Multi-step strategies and allocation. DirRec uses preceding predictions, DIRMO groups output steps (Ben Taieb et al., 2012), boosting corrects horizon-specific errors (Ben Taieb & Hyndman, 2014), and Stratify unifies multi-step strategies (Green et al., 2025b). Generated inputs and multiple predictors are thus established ideas. We combine generated-entrance training with repeated within-stage reuse and adaptive handovers, including sample-specific training-window alignment. Direct/blockwise predictors and shared stage-conditioned networks are alternative ways to allocate horizon-specific capacity; our experiments evaluate separate stagewise predictors.

Our frozen-chain diagnostics distinguish whether a successor helps from whether that advantage can be inferred without future targets.

3 Problem Formulation and Theoretical Analysis

We define stagewise forecasting and state the central specialization, growth, and routing results. Appendices A–C contain auxiliary results and all proofs.

3.1 FORECASTING SETUP AND STAGEWISE TRAINING

Let $X_t \in \mathbb{R}^d$, with history length L, output-block length $1 \le h \le L$, and horizon $H = Nh$. Set $S_0 = X_{t-L+1:t}$ and $Y_j = X_{t+(j-1)h+1:t+jh}$. The update $T(s, y) = \mathrm{tail}_L([s; y])$ appends a block along time and retains the latest L rows. Matrix norms are Frobenius norms. Predictors are deterministic and receive no new observations during rollout; any persistent hidden state must be included in S_j. No known physical equation is required. Future-informed routing is evaluated separately as a diagnostic and lies outside this deployment information set.

Definition 3.1 (Stage partition and rollout risk)**.** A partition $\mathcal{P} = (b_0, \dots, b_K)$ satisfies $0 = b_0 < \cdots < b_K = N$ and assigns calls $I_k = \{b_{k-1} + 1, \dots, b_k\}$ to stage k. Let $\kappa_{\mathcal{P}}(j) = k$ for $j \in I_k$. For predictors $f_{\theta_k} : \mathbb{R}^{L\times d} \to \mathbb{R}^{h\times d}$, the chain $F = (\mathcal{P}, \theta_{1:K})$ produces

$$\widehat{Y}_j = f_{\theta_{\kappa_{\mathcal{P}}(j)}}(S_{j-1}), \qquad S_j = T(S_{j-1}, \widehat{Y}_j). \tag{1}$$

Given fixed weights $w_j > 0$ with $\sum_{j=1}^N w_j = 1$, its population rollout risk is

$$R_{\mathrm{roll}}(F) = \sum_{j=1}^{N} w_j \, \mathbb{E}\|\widehat{Y}_j - Y_j\|^2. \tag{2}$$

Expectations concern a common forecast law conditional on fitted parameters and selected boundaries; subsequent moment and innovation assumptions apply under this law. Within-sequence independence is not required. Dividing risk by hd gives coordinate MSE.

Training on generated contexts. Train stages sequentially. Before fitting stage k, freeze its predecessors and generate entrance states $\widetilde{S}^{(i)}_{b_{k-1}}$ for n training cases. The candidate continues generating contexts within its training interval:

$$U^{(i)}_{b_{k-1}}(\theta) = \widetilde{S}^{(i)}_{b_{k-1}}, \qquad U^{(i)}_j(\theta) = T\Big(U^{(i)}_{j-1}(\theta), f_\theta(U^{(i)}_{j-1}(\theta))\Big), \quad j \in I_k. \tag{3}$$

With $W_k = \sum_{j\in I_k} w_j$, minimize

$$\widehat{\mathcal{L}}_k(\theta) = \frac{1}{nW_k} \sum_{i=1}^{n} \sum_{j\in I_k} w_j \left\| f_\theta(U^{(i)}_{j-1}(\theta)) - Y^{(i)}_j \right\|^2. \tag{4}$$

Generated values supply inputs and true futures supply supervision. Gradients pass through the current unroll with frozen predecessors. Unknown boundaries use a prescribed training interval (Section 4.1). This objective is stagewise, not a joint full-chain fit; caching fixed within-stage contexts changes it (Appendix A.1).

3.2 CONTEXT ADAPTATION AND STAGE REFINEMENT

Generated-context training matches rollout inputs (Proposition A.1); parameter copying permits class refinement (Theorem A.1). Write $\mathcal{P}' \succeq \mathcal{P}$ when a partition only adds boundaries.

Definition 3.2 (Reference risk and stage information)**.** Fix a reference chain $\bar{F}$, and generate $\bar{S}_{j-1}$ from each initial history. Draw J independently of the forecast case with $\Pr(J = j) = w_j$. Set $S = \bar{S}_{J-1}$, $Y = Y_J$, and $G = \kappa_{\mathcal{P}}(J)$. The reference law of (S, Y, J) remains unchanged when comparing partitions or predictors. Define

$$m_{\mathcal{P}} = \mathbb{E}[Y \mid S, G], \qquad B(\mathcal{P}) = \mathbb{E}\|Y - m_{\mathcal{P}}\|^2, \qquad R_{\mathrm{ref}}(u; \mathcal{P}) = \mathbb{E}\|Y - u(S, G)\|^2. \tag{5}$$

Here $B(\mathcal{P})$ minimizes reference risk over all square-integrable measurable predictors of (S, G), not only a chosen neural architecture.

Theorem 3.1 (Exact specialization gain on a common reference law)**.** *For $\mathcal{P}' \succeq \mathcal{P}$ and $\mathbb{E}\|Y\|^2 < \infty$, let $G' = \kappa_{\mathcal{P}'}(J)$. Then*

$$\Delta(\mathcal{P}, \mathcal{P}') := B(\mathcal{P}) - B(\mathcal{P}') = \mathbb{E}\|m_{\mathcal{P}'} - m_{\mathcal{P}}\|^2 \ge 0. \tag{6}$$

The proof is provided in Appendix A.4.

On a fixed reference law, refinement helps when similar contexts require different conditional predictions across stages (Equation (A.4)). This population benchmark does not guarantee gains for fitted networks.

A stage label determined by its context supplies no ideal gain. Fitting error and changed rollout states can offset specialization gains (Corollary A.1, Theorem A.2); greedy fitting need not improve monotonically.

3.3 Horizon-dependent error bounds

For the true states $S_j^\star = T(S_{j-1}^\star, Y_j)$ with $S_0^\star = S_0$, define

$$M_j(F) = \mathbb{E}\|\widehat{Y}_j - Y_j\|^2, \qquad q_j(F) = \left(\mathbb{E}\|S_j - S_j^\star\|^2\right)^{1/2}. \tag{7}$$

M_j is population squared error at block j, not cumulative prefix loss; $M_j/(hd)$ is coordinate MSE.

Theorem 3.2 (Exponential error upper bound for a reused model)**.** *Suppose every call reuses f and its full state update $\Phi(s) = T(s, f(s))$ is γ-Lipschitz between the generated and true states under consideration, with $\gamma > 1$. If the local errors on true contexts satisfy $(\mathbb{E}\|f(S_{j-1}^\star) - Y_j\|^2)^{1/2} \le \varepsilon$ at every call, then, for $1 \le j \le N$,*

$$q_j(F) \le \varepsilon \frac{\gamma^j - 1}{\gamma - 1}, \qquad M_j(F) \le \varepsilon^2 \left(\frac{\gamma^j - 1}{\gamma - 1}\right)^2. \tag{8}$$

If γ, ε hold uniformly over increasing horizons, then $M_j(F) = O(\gamma^{2j})$.

The proof is provided in Appendix A.6.

Definition 3.3 (An innovation component with persistent effects)**.** Let $\mathcal{G}_0 = \sigma(S_0)$ and $\mathcal{G}_i = \sigma(S_0, Y_1, \ldots, Y_i)$. Suppose a fixed matrix $v \in \mathbb{R}^{h \times d}$ of unit Frobenius norm gives a target projection of the form

$$Z_j := \langle v, Y_j \rangle = \mu_j(S_0) + \sum_{i=1}^{j} c_{j,i}\xi_i, \qquad |c_{j,i}| \ge c_* > 0. \tag{9}$$

Here $\langle \cdot, \cdot \rangle$ is the Frobenius inner product, $\mu_j(S_0)$ is square-integrable, and the $c_{j,i}$ are deterministic. Each square-integrable ξ_i is $\mathcal{G}_i$-measurable and satisfies

$$\mathbb{E}[\xi_i \mid \mathcal{G}_{i-1}] = 0, \qquad \mathbb{E}[\xi_i^2 \mid \mathcal{G}_{i-1}] \ge \sigma_*^2 > 0. \tag{10}$$

The constants c_*, σ_* do not depend on i, j over the analyzed horizon.

Theorem 3.3 (Linear error lower bound for switching without new observations)**.** *Under Definition 3.3, every deterministic stagewise chain F with square-integrable predictions and no new true observations during rollout satisfies*

$$M_j(F) \ge \sum_{i=1}^{j} c_{j,i}^2 \mathbb{E}[\xi_i^2] \ge c_*^2 \sigma_*^2 j, \qquad 1 \le j \le N. \tag{11}$$

This holds for any stage count and boundaries, including calls after handovers.

The proof is provided in Appendix A.6.

The lower bound requires non-decaying innovation effects and also applies to one model. It proves neither a switching advantage nor exactly linear realized growth. Fully observed deterministic systems need not meet its assumptions, and the corresponding RMSE floor grows as a square root.

Both bounds concern population expectations, not individual error curves or switching benefits. Future-informed routing (Section 4.2) uses test targets for selection and falls outside Theorem 3.3, even without replacing generated states.

3.4 SWITCHING SIGNALS AND ROUTING RELIABILITY

For a stage entered after a accepted blocks, write r for local age and $j = a + r$ for the global call. Let $\widehat{E}_{k,r}$ estimate coordinate MSE under continued reuse. The two alarm signals are

$$\begin{aligned} H^{\mathrm{lin}}_{k,r} &= \mathbf{1}\{\widehat{E}_{k,r} > \alpha_0 + \beta_0(a+r)\}, \\ H^{\mathrm{slope}}_{k,r} &= \mathbf{1}\{g_{k,r} - \lambda \max(B_{k,r}, s_{\min}) > \zeta\}, \quad g_{k,r} = \widehat{E}_{k,r} - \widehat{E}_{k,r-1}, \end{aligned} \tag{12}$$

Here $B_{k,r}$ is the median of the preceding m increments, excluding the current one; for even windows, average the middle two. Set $\tau(j) = \alpha_0 + \beta_0 j$ and $A_{k,r} = g_{k,r} - \lambda \max(B_{k,r}, s_{\min})$, with $\alpha_0, \beta_0, \zeta \geq 0$, $\lambda > 1$, and $s_{\min} > 0$. Write $A_r(E)$ for this score on an error sequence. After warm-up, p consecutive alarms confirm a handover, subject to the horizon and model budget. Section 4 specifies state retention and deployment.

Proposition B.1 bounds score perturbations by $2(1 + \lambda)\epsilon$ under uniform error control and distinguishes affine from sufficiently fast exponential growth.

For sample-specific decisions, let $e_{k,r} = \|\widehat{Y}_{k,r} - Y_j\|^2/(hd)$. The accepted entrance of stage k is a_k; a deployable policy replaces unavailable future errors with conditional risk.

Definition 3.4 (Deployable error-prediction query)**.** Before accepting a candidate block at global position $j = a_k + r$, form

$$Z_{k,r} = \psi(S_0, S_{j-1}, \widehat{Y}_{k,r}, j, r), \qquad \mu_k(z) = \mathbb{E}[e_{k,r} \mid Z_{k,r} = z]. \tag{13}$$

The feature map ψ uses only initial observations, generated states, the tentative forecast, and position information. The conditional law is that of candidate calls from the specified deployable prefix, including diagnostic continuation of the current model. The error predictor $q_{\phi_k}(z) \geq 0$ estimates this conditional mean on its query domain.

The following guarantee concerns the linear and slope rules with conditional mean risks; it does not describe the realized-error comparison introduced later.

Proposition 3.1 (Routing agreement and transfer of loss quantiles)**.** *Fix the fitted forecasters, feature map, switching parameters, and model budget. Let Π_μ apply the inference procedure with μ_k, and let Π_q use q_{ϕ_k}. Take $\epsilon \geq 0$ and $0 \leq \delta < 1/2$. On an event $\mathcal{G}$ with probability at least $1 - \delta$ over a new sample and any independent internal sampling randomness, assume the following along every candidate call of Π_μ, including warm-up and discarded probes: $|q_{\phi_k}(Z_{k,r}) - \mu_k(Z_{k,r})| \leq \epsilon$. At each monitored call, also assume the margin for the chosen rule:*

$$\begin{cases} |\mu_k(Z_{k,r}) - \tau(j)| > \epsilon, & \textit{linear rule}, \\ |A_{k,r}(\mu) - \zeta| > 2(1+\lambda)\epsilon, & \textit{slope rule}. \end{cases} \tag{14}$$

Here $A_{k,r}(\mu)$ uses the within-stage conditional-risk sequence. Then both procedures have identical accepted forecasts on $\mathcal{G}$. For their full-horizon sample losses $L_\Pi = \sum_{j=1}^N w_j \|\widehat{Y}^\Pi_j - Y_j\|^2/(hd)$, write $F_\Pi(x) = \Pr(L_\Pi \leq x)$ and $Q_\eta(L) = \inf\{x : \Pr(L \leq x) \geq \eta\}$. For $\delta < \eta < 1 - \delta$,

$$\begin{aligned} &\sup_x |F_{\Pi_q}(x) - F_{\Pi_\mu}(x)| \leq \delta, \\ &Q_{\eta-\delta}(L_{\Pi_\mu}) \;\leq\; Q_\eta(L_{\Pi_q}) \leq Q_{\eta+\delta}(L_{\Pi_\mu}). \end{aligned} \tag{15}$$

The proof is provided in Appendix C.5.

Average calibration does not establish these uniform premises. Exact conditional risks need not recover realized errors, and routing agreement does not ensure an effective rule. A poorly chosen rule can fail even with exact risks; prediction and routing shifts can add errors. Chain selection is analyzed in Theorem B.1.

4 METHODOLOGY: TWO ADAPTIVE STRATEGIES

Figure 1 contrasts shared public boundaries with individual handovers and routing. Both methods train on generated contexts using the signals in Section 3.4.

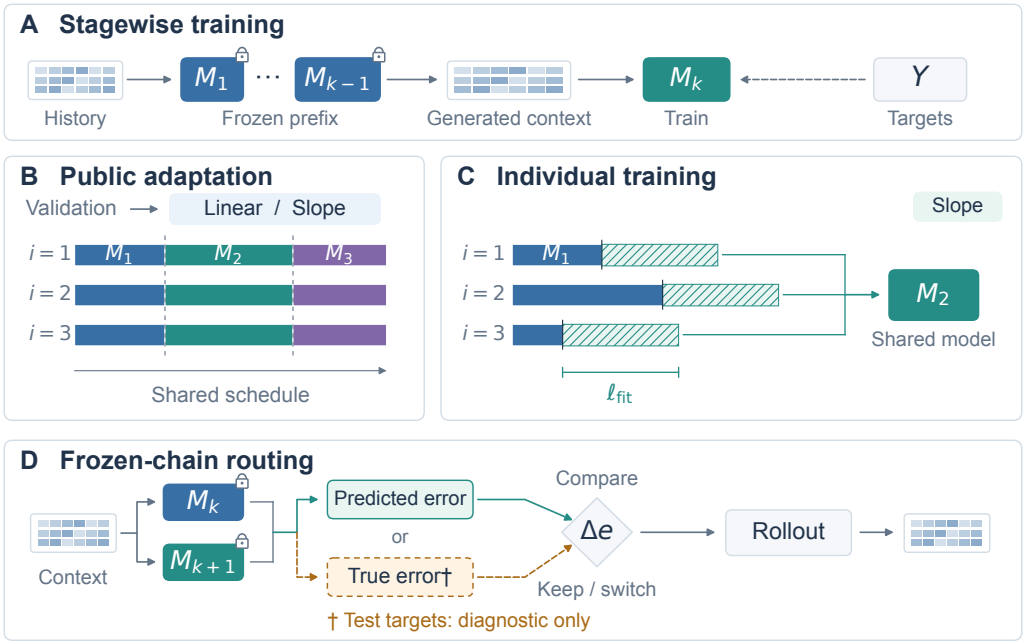


Figure 1: **Framework overview.** (A) Frozen models generate contexts. (B) Validation selects shared boundaries. (C) Individual handovers align fitting windows for a shared successor. (D) Freeze the chain and compare adjacent models. Orange marks diagnostic future-target access; Δe is relative sRMSE reduction, with $\rho = 0.05$ in (19). Intervals are schematic.

4.1 Public adaptation from error levels and slopes

Training and boundary construction. Separate forecaster-training ($\mathcal{D}_{\text{tr}}$), boundary-construction ($\mathcal{D}_{\text{b}}$), and test sequences; fit normalization on training data. Optional chain selection uses an additional independent set $\mathcal{D}_{\text{s}}$. At accepted position $a = b_{k-1}$, generate entrance states S_a with the finalized frozen prefix. Train model k on $\widetilde{I}_k = \{a+1, \ldots, \min(a+\ell_{\text{fit}}, N)\}$ using (4), renormalizing weights and differentiating through its own unroll. This prescribed interval avoids dependence on an unknown boundary; longer reuse is monitored extrapolation. Freeze the model before probing construction cases. At local age r and global position $j = a + r$, measure per-block coordinate MSE:

$$\widehat{E}_{k,r} = \frac{1}{n_{\text{b}} h d} \sum_{i=1}^{n_{\text{b}}} \|\widehat{Y}_{k,r}^{(i)} - Y_{a+r}^{(i)}\|^2. \tag{16}$$

Here $n_{\text{b}} = |\mathcal{D}_{\text{b}}|$. Targets score predictions without replacing generated states.

Monitoring and handovers. The linear rule compares error with $\alpha_0 + \beta_0 j$, retaining global time across handovers. Theorem 3.3 motivates this scale but supplies neither an upper envelope nor threshold coefficients. The slope rule compares the current increment with a multiple of the preceding increments' median, using a positive floor and slack. It detects acceleration; high flat errors can escape detection, and alarms do not uniquely identify exponential growth. Calibrate both policies on $\mathcal{D}_{\text{b}}$.

After $r_0 = m+1$ warm-up calls, with $\ell_{\text{fit}} \geq r_0$, require p consecutive alarms. On confirmation with budget remaining, set $b_k = a + r - 1$, discard the current probe, and let the successor recompute it from the retained state. Earlier accepted outputs remain. Regenerate training entrances with the finalized prefix and reset monitoring at handover. Terminal handling and counter resets are specified in Appendix B.

Chain selection and deployment. Deployment uses shared boundaries, K stored models and N calls, without future errors. Experiments compare the controllers directly. Optional independent selection can minimize model count within a dense-baseline tolerance (Theorem B.1); its bounded-loss guarantee is not claimed for these experimental results.

4.2 Sample-specific training and comparative routing

Train individual entrances, then freeze the chain. Let $a_k^{(i)}$ be sample i's accepted entrance into shared model k. The first model has $a_1^{(i)} = 0$; subsequent supervised windows are

$$\widetilde{I}_k^{(i)} = \{a_k^{(i)} + 1, \ldots, \min(a_k^{(i)} + \ell_{\text{fit}}, N)\}. \tag{17}$$

With active cases $\mathcal{A}_k = \{i : a_k^{(i)} < N\}$, fit one shared successor:

$$\widehat{\theta}_k \in \arg\min_{\theta} \frac{1}{|\mathcal{A}_k|} \sum_{i \in \mathcal{A}_k} \frac{1}{W_k^{(i)}} \sum_{j \in \widetilde{I}_k^{(i)}} w_j \| f_\theta(U_{j-1}^{(i)}(\theta)) - Y_j^{(i)} \|^2. \tag{18}$$

Here $W_k^{(i)} = \sum_{j \in \widetilde{I}_k^{(i)}} w_j$ and $U_j^{(i)}(\theta)$ follows (3) from generated entrances, with predecessors frozen. Training uses each case's realized error and the slope policy to determine handovers. Confirmation at local probe r sets $a_{k+1}^{(i)} = a_k^{(i)} + r - 1$; the successor learns its next window, not a common absolute interval. Training targets never replace generated states.

All sample policies use this frozen slope-trained chain; only inference changes. Different routes may therefore encounter entrances absent during training.

Why individual level and slope rules can fail. Population bounds do not prescribe individual error curves, and current-model deterioration does not establish successor superiority. Even an exact-error alarm may prompt an unhelpful handover; adjacent comparison directly tests successor advantage.

Adjacent-model comparison with privileged errors. For each sample, evaluate adjacent models on the same generated context and target interval, with identical channel scales. Let $\ell_{k,j}$ and $\ell_{k+1,j}$ denote their sRMSE values. Switch when the successor reduces sRMSE by at least $\rho = 0.05$:

$$H_{k,j}^{\text{cmp}} = \mathbf{1}\{\ell_{k,j} > 0,\ \ell_{k+1,j} \le (1-\rho)\ell_{k,j}\}. \tag{19}$$

Zero-error ties retain the current model. This TrueCmp diagnostic uses future targets for selection, retaining generated states. It is neither deployable nor globally optimal: local choices alter later contexts and opportunities. The 5% threshold applies to sRMSE, not MSE; interval and acceptance details appear in Appendix C.4.

Learned routing and its limitations. PredCmp replaces true sRMSE with estimates from observed/generated histories, candidate outputs, and positions. Its sRMSE interface adapts Definition 3.4; fitting and calibration use separate sequences without test targets (Appendix C.2).

The MAST predictors do not recover TrueCmp's gains. Residual heterogeneity, near-margin ranking, and changed contexts may contribute; exact conditional means need not recover realized errors. Privileged and learned routing are evaluated separately; individual routing remains unresolved.

5 Experiments

5.1 Datasets, backbones, and evaluation

Datasets and backbones. We forecast TokaMark-curated **MAST** tokamak signals (Rousseau et al., 2026; Jackson et al., 2024), extended **TEP** chemical-process trajectories (Reinartz et al., 2021), and **C-MAPSS** turbofan degradation sequences (Saxena & Goebel, 2008). TFT (Lim et al., 2021), DAG (Qiu et al., 2026), and Timer-XL (Liu et al., 2025) share available inputs and horizons within each dataset. Covariates must be known at forecast origin; future targets are accessed only by marked routing diagnostics. Appendix D.1 describes sources and metrics.

Table 1: Fixed-stage results. R/G: observed/generated contexts. $\bar{r}_{500}$: mean per-sample sRMSE at step 500; T_{fit}: training including context generation (hours:minutes). Bold marks lower paired error.

Dataset	Backbone	Input	$K = 5$		$K = 10$		$K = 20$	
			$\bar{r}_{500}$	T_{fit}	$\bar{r}_{500}$	T_{fit}	$\bar{r}_{500}$	T_{fit}
MAST	TFT	R	0.06971	2:33	0.05815	3:59	0.06391	4:31
		G	**0.03739**	4:33	**0.03726**	8:37	**0.03546**	15:15
	DAG	R	0.48148	1:50	0.77653	2:53	0.69373	3:44
		G	**0.07758**	3:25	**0.07074**	6:04	**0.06499**	10:53
	Timer-XL	R	0.08574	1:17	0.05035	2:05	0.05177	3:57
		G	**0.03805**	1:58	**0.03596**	3:39	**0.03745**	7:58
TEP	TFT	R	0.28306	3:35	0.24065	5:11	0.28433	6:24
		G	**0.15295**	6:21	**0.11991**	11:57	**0.11024**	18:55
	DAG	R	0.37929	2:05	1.02337	3:16	2.01107	8:37
		G	**0.28140**	3:35	**0.26029**	6:38	**0.20399**	11:45
	Timer-XL	R	0.17727	1:23	0.14425	2:11	0.14304	3:12
		G	**0.10982**	1:52	**0.09546**	3:25	**0.09179**	5:54
C-MAPSS	TFT	R	0.01792	4:30	0.01772	6:55	0.01718	7:27
		G	**0.01303**	7:44	**0.00819**	14:32	**0.00476**	25:05
	DAG	R	0.15864	3:14	0.16413	5:07	0.17124	5:33
		G	**0.01826**	5:53	**0.01279**	10:56	**0.00644**	18:57
	Timer-XL	R	0.00482	2:51	0.00512	4:28	0.00611	4:44
		G	**0.00397**	4:49	**0.00229**	8:52	**0.00197**	15:01

Table 2: Adaptive results: endpoint sRMSE mean ($\bar{r}$) and $q90$/$q95$ at lead 500; mean used models ($\bar{K}$). (a) Fixed-K, public Linear and Slope policies on three datasets. (b) MAST-only sample routing; fixed baselines are in (a). Bold: best deployable errors per dataset/backbone; adaptive model counts. †: future-target diagnosis, **not deployable**. Tradeoffs: Table D.5.

(a) Fixed stages and public adaptation

Method	MAST				TEP				C-MAPSS			
	$\bar{r}$	$q90$	$q95$	$\bar{K}$	$\bar{r}$	$q90$	$q95$	$\bar{K}$	$\bar{r}$	$q90$	$q95$	$\bar{K}$
TFT												
Fixed-5	0.03739	0.07063	**0.09983**	5	0.15295	0.29173	0.40731	5	0.01303	0.01970	0.02403	5
Fixed-10	0.03726	0.07684	0.11338	10	0.11991	0.20443	0.29957	10	0.00819	0.01248	0.01544	10
Fixed-20	0.03546	**0.06589**	0.10187	20	**0.11024**	**0.18602**	**0.26974**	20	**0.00476**	**0.00727**	**0.00943**	20
Linear	**0.03528**	0.07131	0.10500	**9.0**	0.13048	0.23553	0.33476	**9.0**	0.00829	0.01419	0.01891	**8.0**
Slope	0.03832	0.07420	0.10366	**6.0**	0.22977	0.46981	0.67294	**3.0**	0.00774	0.01250	0.01548	**8.0**
DAG												
Fixed-5	0.07758	0.13176	0.15995	5	0.28140	0.61869	0.93162	5	0.01826	0.02786	0.03230	5
Fixed-10	0.07074	0.12175	0.15117	10	0.26029	0.54304	0.74459	10	0.01279	0.01920	0.02408	10
Fixed-20	**0.06499**	**0.11674**	**0.14792**	20	**0.20399**	**0.41924**	**0.56924**	20	**0.00644**	**0.01119**	**0.01272**	20
Linear	0.07074	0.12175	0.15117	**10.0**	0.25930	0.58637	0.81910	**10.0**	0.01132	0.01780	0.02008	**5.0**
Slope	0.06837	0.12122	0.14811	**7.0**	0.40345	0.75756	1.10521	**3.0**	0.13786	0.18796	0.20186	**1.0**
Timer-XL												
Fixed-5	0.03805	0.06594	0.09221	5	0.10982	0.18870	0.25537	5	0.00397	0.00611	0.00849	5
Fixed-10	0.03596	0.06454	0.09175	10	0.09546	0.16000	0.21287	10	0.00229	0.00395	0.00493	10
Fixed-20	0.03745	0.07357	0.12096	20	**0.09179**	**0.14640**	**0.20479**	20	**0.00197**	**0.00283**	**0.00374**	20
Linear	0.03572	0.06552	0.09261	**8.0**	0.10796	0.18523	0.24920	**6.0**	0.00277	0.00427	0.00577	**4.0**
Slope	**0.03528**	**0.06442**	**0.09020**	**7.0**	0.09723	0.16126	0.21214	**8.0**	0.00215	0.00340	0.00476	**8.0**

(b) Sample-specific routing on MAST

Backbone	Sample–PredCmp				Sample–TrueCmp†			
	$\bar{r}$	$q90$	$q95$	$\bar{K}$	$\bar{r}$	$q90$	$q95$	$\bar{K}$
TFT	0.03774	0.07481	0.10606	**6.7**	0.03113	0.05765	0.08668	**5.7**
DAG	0.09230	0.15180	0.18675	**6.8**	0.06977	0.11935	0.14371	**7.1**
Timer-XL	0.03630	0.06977	0.09997	**7.0**	0.03269	0.05648	0.07914	**5.6**

Metrics and protocol. All tabulated errors concern forecast lead 500: mean sRMSE averages per-sample endpoint errors, and $q90$/$q95$ are their across-case quantiles. Table 1 reports training time including context generation; Table 2 reports mean model use. Adaptive latency is unmeasured; metric definitions appear in Appendix D.1.

5.2 Observed versus generated training contexts

With true-future supervision and autonomous test rollouts, generated (G) versus observed (R) contexts reduce $\bar{r}_{500}$ by 17.63%–96.24% in all 27 matched comparisons at $K \in \{5, 10, 20\}$ (Table 1). Training takes longer in every pair; model counts, not training times, are matched.

5.3 Adaptive handovers: accuracy, tail error, and cost

Table 2(a) compares fixed/public policies; (b) isolates MAST routing. On MAST, slope control uses $6/7/7$ models for TFT/DAG/Timer-XL versus ten, with mean-error changes of $+2.82\%/-3.35\%/-1.89\%$. TrueCmp improves all three errors over PredCmp on every backbone. It is best on TFT/Timer-XL; DAG fixed-20 has lower mean and $q90$, while TrueCmp has lower $q95$.

On C-MAPSS, TFT/Timer-XL slope and DAG linear control improve mean error with fewer models, but DAG slope yields $10.78\times$ the ten-model error. On TEP, TFT/DAG slope use three models but raise mean error by $91.63\%/55.00\%$; Timer-XL slope uses eight with a 1.86% increase. Neither controller preserves accuracy universally. Appendix D.5 reports every controller's mean/tail tradeoff.

Mean use $\bar{K} = n^{-1} \sum_i K_i$ counts distinct forecasters used in accepted forecasts, excluding rejected probes; it does not establish runtime savings. Appendix D.6 analyzes the MAST routing gap. Future-informed results do not establish deployable superiority.

5.4 Model-count scaling on C-MAPSS

C-MAPSS improves at each stage count (Figure 2); from 10 to 20, TFT/DAG/Timer-XL errors fall by $41.88\%/49.65\%/13.97\%$, with higher training cost. In contrast, MAST/Timer-XL worsens by 4.15%. Cross-dataset changes and the separate dense MAST sweep appear in Appendix D.3.

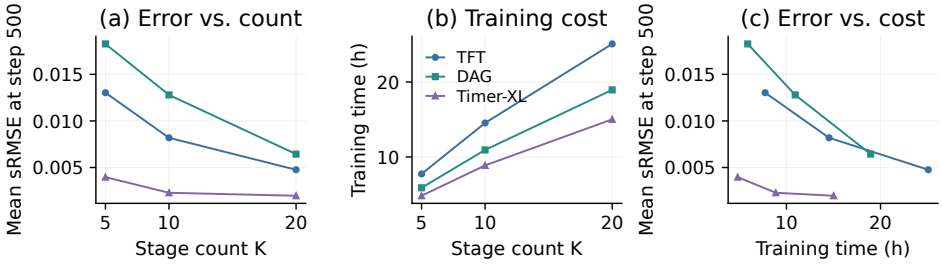


Figure 2: C-MAPSS generated-context scaling at $K = 5, 10, 20$: mean sRMSE at step 500 and training time. Lines join measured points; no adaptive results are inferred.

6 Conclusion

Generated-context training improves all 27 matched comparisons across three datasets and backbones. Scaling remains system- and backbone-dependent: C-MAPSS improves steadily, whereas MAST shows smaller gains or deterioration as training costs rise. Public adaptation can retain comparable accuracy with substantially fewer models: on MAST/DAG, seven models incur only 5.20% higher mean sRMSE than twenty, a 65% model-count reduction. This tradeoff remains configuration-dependent, with substantial failures elsewhere. On MAST, future-informed adjacent-model comparison improves endpoint mean and tail errors over learned routing, without dominating every fixed baseline. Its advantage remains diagnostic; reliable deployment requires better learned decisions. Limitations are discussed in Appendix E.

AI USE STATEMENT

Generative AI tools were used to assist with manuscript writing. The authors are responsible for reviewing and verifying all AI-assisted content and take full responsibility for the final manuscript.

REPRODUCIBILITY STATEMENT

Sections 3–4 specify the forecasting setup, training objectives, and switching rules; Appendices A–C provide assumptions, proofs, and algorithmic details. Dataset sources and backbone references are given in Section 5, with supplementary comparisons in Appendix D. The learned and true-error comparison policies use the same frozen model chain and a common 5% sRMSE switching threshold (Section 4.2). Deployable policies are explicitly distinguished from diagnostics that access future test targets.

ETHICS STATEMENT

This work studies forecasting on existing scientific and engineering datasets. Average benchmark accuracy does not establish reliability for individual trajectories or under distribution shift. Forecasts used in consequential physical-system decisions require application-specific validation and monitoring. Future-target-assisted routing is reported solely as a diagnostic, not as a deployable forecasting policy.

REFERENCES

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Yuyang Wang. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=gerNCVqqtR.

Souhaib Ben Taieb and Rob J. Hyndman. Boosting multi-step autoregressive forecasts. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pp. 109–117. PMLR, 2014. URL https://proceedings.mlr.press/v32/taieb14.html.

Souhaib Ben Taieb, Gianluca Bontempi, Amir F. Atiya, and Antti Sorjamaa. A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition. *Expert Systems with Applications*, 39(8):7067–7083, 2012. doi: 10.1016/j.eswa.2012.01.039.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, volume 28, 2015. URL https://arxiv.org/abs/1506.03099.

Gavin C. Cawley and Nicola L. C. Talbot. On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research*, 11:2079–2107, 2010. URL https://jmlr.org/papers/v11/cawley10a.html.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 10148–10167. PMLR, 2024. URL https://proceedings.mlr.press/v235/das24c.html.

Tilmann Gneiting. Making and evaluating point forecasts. *Journal of the American Statistical Association*, 106(494):746–762, 2011. doi: 10.1198/jasa.2011.r10138.

Riku Green, Huw Day, Zahraa S. Abdallah, and Telmo M. Silva Filho. Epistemic error decomposition for multi-step time series forecasting: Rethinking bias-variance in recursive and direct strategies. *arXiv preprint arXiv:2511.11461*, 2025a. URL https://arxiv.org/abs/2511.11461v1. Presented at the 1st Workshop on Epistemic Intelligence in Machine Learning.

Riku Green, Grant Stevens, Zahraa S. Abdallah, and Telmo M. Silva Filho. Stratify: unifying multi-step forecasting strategies. *Data Mining and Knowledge Discovery*, 39(5):64, 2025b. doi: 10.1007/s10618-025-01135-1.

Ernst Hairer, Syvert P. Nørsett, and Gerhard Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems*, volume 8 of *Springer Series in Computational Mathematics*. Springer, 2 edition, 1993. doi: 10.1007/978-3-540-78862-1.

Xizewen Han, Huangjie Zheng, and Mingyuan Zhou. CARD: Classification and regression diffusion models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 18100–18115. Curran Associates, Inc., 2022. doi: 10.52202/068431-1316.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851. Curran Associates, Inc., 2020. URL https://arxiv.org/abs/2006.11239.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. doi: 10.1080/01621459.1963.10500830.

Rob J. Hyndman and George Athanasopoulos. *Forecasting: Principles and Practice*. OTexts, Melbourne, Australia, 3 edition, 2021. URL https://otexts.com/fpp3/.

Samuel Jackson, Saiful Khan, Nathan Cummings, James Hodson, Shaun de Witt, Stanislas Pamela, Rob Akers, Jeyan Thiyagalingam, and The MAST Team. FAIR-MAST: A fusion device data management system. *SoftwareX*, 27:101869, 2024. doi: 10.1016/j.softx.2024.101869. URL https://doi.org/10.1016/j.softx.2024.101869.

Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, Alexander Merose, Stephan Hoyer, George Holland, Oriol Vinyals, Jacklynn Stott, Alexander Pritzel, Shakir Mohamed, and Peter Battaglia. Learning skillful medium-range global weather forecasting. *Science*, 382(6677):1416–1421, 2023. doi: 10.1126/science.adi2336.

Alex M. Lamb, Anirudh Goyal, Ying Zhang, Saizheng Zhang, Aaron C. Courville, and Yoshua Bengio. Professor forcing: A new algorithm for training recurrent networks. In *Advances in Neural Information Processing Systems*, volume 29, 2016. URL https://arxiv.org/abs/1610.09038.

Zheng Li, Jerry Cheng, and Huanying Helen Gu. AROpt: An optimization method for autoregressive time series forecasting. *arXiv preprint arXiv:2602.02288*, 2026. URL https://arxiv.org/abs/2602.02288v3.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2010.08895.

Bryan Lim, Sercan Ö. Arık, Nicolas Loeff, and Tomas Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4):1748–1764, 2021. doi: 10.1016/j.ijforecast.2021.03.012.

Phillip Lippe, Bas Veeling, Paris Perdikaris, Richard Turner, and Johannes Brandstetter. PDE-Refiner: Achieving accurate long rollouts with neural PDE solvers. In *Advances in Neural Information Processing Systems*, volume 36, pp. 67398–67433, 2023. doi: 10.52202/075280-2946. URL https://arxiv.org/abs/2308.05732.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. iTransformer: Inverted transformers are effective for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024a. URL https://openreview.net/forum?id=JePfAI8fah.

Yong Liu, Haoran Zhang, Chenyu Li, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer: Generative pre-trained transformers are large time series models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 32369–32399. PMLR, 2024b. URL https://proceedings.mlr.press/v235/liu24cb.html.

Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer-XL: Long-context transformers for unified time series forecasting. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.04803.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Jbdc0vT0col.

Boris N. Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/1905.10437.

Jaideep Pathak, Brian Hunt, Michelle Girvan, Zhixin Lu, and Edward Ott. Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach. *Physical Review Letters*, 120(2):024102, 2018. doi: 10.1103/PhysRevLett.120.024102.

Xiangfei Qiu, Yuhan Zhu, Zhengyu Li, Xingjian Wu, Bin Yang, and Jilin Hu. DAG: A dual correlation network for time series forecasting with exogenous variables. In *International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/62103.

Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8857–8868. PMLR, 2021. URL https://proceedings.mlr.press/v139/rasul21a.html.

Christopher Clarc Reinartz, Murat Kulahci, and Ole Ravn. An extended Tennessee Eastman simulation dataset for fault-detection and decision support systems. *Computers & Chemical Engineering*, 149:107281, 2021. doi: 10.1016/j.compchemeng.2021.107281. URL https://doi.org/10.1016/j.compchemeng.2021.107281.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 627–635. PMLR, 2011. URL https://proceedings.mlr.press/v15/ross11a.html.

Cécile Rousseau, Samuel Jackson, Rodrigo H. Ordonez-Hurtado, Nicola C. Amorisco, Tobia Boschi, George K. Holt, Andrea Loreti, Eszter Székely, Alexander Whittle, Adriano Agnello, Stanislas Pamela, Alessandra Pascale, Robert Akers, Juan Bernabe Moreno, Sue Thorne, and Mykhaylo Zayats. TokaMark: A comprehensive benchmark for MAST tokamak plasma models. *arXiv preprint arXiv:2602.10132*, 2026. doi: 10.48550/arXiv.2602.10132. URL https://arxiv.org/abs/2602.10132.

David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3): 1181–1191, 2020. doi: 10.1016/j.ijforecast.2019.07.001.

Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8459–8468. PMLR, 2020. URL https://proceedings.mlr.press/v119/sanchez-gonzalez20a.html.

Abhinav Saxena and Kai Goebel. Turbofan engine degradation simulation data set. NASA Prognostics Data Repository, NASA Ames Research Center, 2008. URL https://www.nasa.gov/intelligent-systems-division/discovery-and-systems-health/pcoe/pcoe-data-set-repository/.

Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen, and Ming Jin. Time-MoE: Billion-scale time series foundation models with mixture of experts. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=e1wDDFmlVu.

Natasa Tagasovska and David Lopez-Paz. Single-model uncertainties for deep learning. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://arxiv.org/abs/1811.00908.

Philipp Teutsch and Patrick Mäder. Flipped classroom: Effective teaching for time series forecasting. *Transactions on Machine Learning Research*, 2022. URL https://openreview.net/forum?id=w3x20YEcQK.

Charles Truong, Laurent Oudre, and Nicolas Vayatis. Selective review of offline change point detection methods. *Signal Processing*, 167:107299, 2020. doi: 10.1016/j.sigpro.2019.107299.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 53140–53164. PMLR, 2024. URL https://proceedings.mlr.press/v235/woo24a.html.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Advances in Neural Information Processing Systems*, volume 34, pp. 22419–22430, 2021. URL https://arxiv.org/abs/2106.13008.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 11121–11128, 2023. doi: 10.1609/aaai.v37i9.26317.

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. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 11106–11115, 2021. doi: 10.1609/aaai.v35i12.17325.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 27268–27286. PMLR, 2022. URL https://proceedings.mlr.press/v162/zhou22g.html.

A STAGEWISE FORMULATION AND PROOFS

We formalize stagewise training and distinguish frozen-reference risk from actual rollout risk. Multi-output and hybrid strategies already allocate parameters across prediction horizons (Ben Taieb et al., 2012; Green et al., 2025b); here we analyze predictor reuse within stages. The main text retains the central results; auxiliary statements and all stagewise proofs appear below. We use standard conditional-mean identities for squared loss (Gneiting, 2011) to connect context adaptation, specialization, and rollout stability.

A.1 FORECASTING STATES AND SUPERVISED STAGEWISE TRAINING

Let $X_t \in \mathbb{R}^d$, with history length L and output-block length $1 \le h \le L$. For clarity, the forecast horizon is $H = Nh$; a non-divisible horizon can be handled by masking the unused entries of the last block. The analysis below uses complete blocks. At a forecast origin t, write $S_0 = X_{t-L+1:t}$ and $Y_j = X_{t+(j-1)h+1:t+jh}$. All matrix norms are Frobenius norms, denoted by $\|\cdot\|$. The state update is $T(s, y) = \mathrm{tail}_L([s; y])$, where concatenation is along time. We consider deterministic predictors with no hidden state carried outside S_j and no new observations during a rollout; any additional persistent state must be included in the state definition.

The definition is stated in Definition 3.1.

Expectations are over forecast cases from a common population law. Population comparisons condition on fitted parameters and chosen boundaries; subsequent moment and innovation assumptions must hold under this conditional forecast law. No independence between positions within a sequence is required. Uniform weights give the average block loss; division by hd gives per-coordinate mean squared error.

Training at the entrance and within a stage. For a fixed partition, train stages in order. After fitting stages $1, \dots, k-1$, freeze them and generate the entrance state $\widetilde{S}^{(i)}_{b_{k-1}}$ for training cases $i = 1, \dots, n$ from their initial histories. The candidate-dependent states within stage k follow equation (3). With $W_k = \sum_{j \in I_k} w_j$, equation (4) defines the stage objective. Equation (4) assumes a known stage interval; adaptive construction in Appendix B instead uses a prescribed training unroll and separately selects the reuse endpoint. Generated values supply inputs; targets remain observed future blocks from training sequences. Gradients pass through the selected training unroll, while the frozen prefix contributes no parameter gradients. This supervises multiple calls, not only the first at a handover. Holding cached U_j fixed during fitting gives a different surrogate; entrance matching alone does not establish adaptation throughout the deployed stage.

Training order and inference. At each stage, initialize the predictor, approximately minimize (4), and freeze the selected parameters before constructing the next stage. At inference, execute (1) and preserve the generated state at every handover. This greedy procedure does not globally minimize (2): changing an early stage can affect every later input. Validation-based boundary selection is addressed in Appendix B; validation cases are not used as supervised training targets. At fixed H and h, the chain accepts N forecast blocks; discarded probes are counted separately. The number of stage parameter sets, training cost, and inference cost are separate quantities.

A.2 WHY GENERATED CONTEXTS CHANGE THE PREDICTION TASK

Fix a stage and context-generation mechanism. Let P, Q be the joint laws of observed or generated contexts and aligned targets, with context marginals P_S, Q_S. Write $R_Q(f) = \mathbb{E}_Q\|Y - f(S)\|^2$ and $m_Q(S) = \mathbb{E}_Q[Y \mid S]$.

Proposition A.1 (Risk matching on generated contexts)**.** *Assume $\mathbb{E}_Q\|Y\|^2 < \infty$ and $f(S) \in L^2(Q)$. Then*

$$R_Q(f) = R_Q(m_Q) + \mathbb{E}_Q\|f(S) - m_Q(S)\|^2. \tag{A.1}$$

In particular, let $m_P(S) = \mathbb{E}_P[Y \mid S]$, assume $\mathbb{E}_P\|Y\|^2 < \infty$, $Q_S \ll P_S$, and $m_P \in L^2(Q_S)$. The excess risk of the observed-context Bayes predictor on generated contexts is

$$R_Q(m_P) - R_Q(m_Q) = \mathbb{E}_Q\|m_P(S) - m_Q(S)\|^2. \tag{A.2}$$

Proof of Proposition A.1.

Proof. Expand $Y - f(S) = (Y - m_Q(S)) + (m_Q(S) - f(S))$. The cross term has expectation zero because $\mathbb{E}_Q[Y - m_Q(S) \mid S] = 0$ and $m_Q(S) - f(S)$ is measurable with respect to S. Square integrability makes this product integrable, giving (A.1). Absolute continuity $Q_S \ll P_S$ makes the P_S-almost-sure definition of m_P unambiguous on Q_S; substituting $f = m_P$ gives (A.2). □

The mismatch is strictly costly exactly when the conditional means differ on a set of positive Q_S measure. A change in input marginals alone need not create this cost. Without support coverage, m_P requires an explicitly chosen extrapolation on generated inputs; the first identity still applies to any square-integrable fitted predictor, but P alone does not determine its extrapolation. This result compares ideal predictors on a fixed Q, not the outcomes of finite-sample non-convex training. In (4), changing the current predictor also changes the within-stage context law.

A.3 What increasing the number of stages guarantees

Let $\mathcal{A}(\mathcal{P})$ contain the complete mappings $S_0 \mapsto (\widehat{Y}_1, \ldots, \widehat{Y}_N)$ realizable by $\mathcal{P}$. A refinement $\mathcal{P}' \succeq \mathcal{P}$ adds boundaries without removing old ones.

Theorem A.1 (Refinement of realizable rollout chains)**.** *Suppose every child stage can use any predictor allowed for its parent, including an exact parameter copy, and handovers neither reset nor otherwise alter the state. Then*

$$\mathcal{A}(\mathcal{P}) \subseteq \mathcal{A}(\mathcal{P}'), \qquad \inf_{F \in \mathcal{A}(\mathcal{P}')} R_{\mathrm{roll}}(F) \leq \inf_{F \in \mathcal{A}(\mathcal{P})} R_{\mathrm{roll}}(F). \tag{A.3}$$

If every nonempty contiguous partition with K stages is admissible, the optimal rollout risk with $K+1$ parameter slots cannot exceed that with K slots, for $K < N$.

Proof of Theorem A.1.

Proof. Take any chain in $\mathcal{A}(\mathcal{P})$ and copy each parent predictor into all of its child stages. Both chains start from the same S_0. If their states agree before call j, they apply identical predictors, produce identical blocks, and obtain identical next states under T. Induction proves equality of the complete forecast for every initial state. This establishes inclusion and the infimum inequality. For $K < N$, at least one stage contains two or more calls and can be split, giving the final claim by taking the union over admissible partitions. □

This is a statement about the best realizable chain, including its own generated inputs. It does not require frozen reference trajectories, attainment of an infimum, or successful global optimization. It also does not assert strict improvement. Fixed total parameter budgets, mandatory changes in backbone size, or independently imposed equal-length partitions can invalidate the inclusion. In particular, greedy stagewise fitting need not find the embedded copy or the population optimum.

A.4 Quantifying specialization and its saturation

To quantify a split's potential benefit, we now freeze the context distribution. Related multi-step analyses distinguish approximation and estimation effects (Ben Taieb & Hyndman, 2014; Green et al., 2025a); here the reference law lets us isolate the information supplied by stage labels.

The definition is stated in Definition 3.2.

Single-stage split. If a single parent stage g is split into children g_1, g_2, write $\pi(s) = \Pr(G' = g_1 \mid S = s, G = g)$ and $\mu_a(s) = \mathbb{E}[Y \mid S = s, G' = g_a]$. Then

$$\Delta(\mathcal{P}, \mathcal{P}') = \mathbb{E}\big[\mathbf{1}_{\{G=g\}} \pi(S)(1 - \pi(S)) \|\mu_1(S) - \mu_2(S)\|^2\big]. \tag{A.4}$$

Values of a child mean where that child has zero conditional probability are immaterial.

Proof of Theorem 3.1.

Proof. Refinement makes G a function of G', so $\sigma(S, G) \subseteq \sigma(S, G')$ and the tower property gives $\mathbb{E}[m_{\mathcal{P}'} \mid S, G] = m_{\mathcal{P}}$. Expand $Y - m_{\mathcal{P}} = (Y - m_{\mathcal{P}'}) + (m_{\mathcal{P}'} - m_{\mathcal{P}})$. Conditioning on (S, G') makes the cross term zero, proving (6). For the split parent, its conditional mean is $\mu = \pi\mu_1 + (1-\pi)\mu_2$. Conditional on $(S, G = g)$, the squared mean difference averages to

$$\pi\|\mu_1 - \mu\|^2 + (1-\pi)\|\mu_2 - \mu\|^2 = \pi(1-\pi)\|\mu_1 - \mu_2\|^2.$$

Other stages are unchanged. Averaging proves (A.4). □

The two factors in (A.4) identify when a split is useful: child stages must require different conditional means, and the context alone must not already identify the child. If G' is measurable from (S, G), the gain is zero. If $\|m_{\mathcal{P}'} - m_{\mathcal{P}}\| \leq \varepsilon$ almost surely, the gain is at most ε^2. These are conditional saturation criteria; the theorem does not imply that gains decrease after every split or that a universal finite stage count reaches a plateau. A sufficiently expressive shared predictor given the stage label can realize the same conditional mean, so this gain does not by itself establish a need for independent networks.

Corollary A.1 (Learned error and the value of an additional stage)**.** *For fixed fitted predictors $u_{\mathcal{P}}$ and $u_{\mathcal{P}'}$, define $A_{\mathcal{P}} = \mathbb{E}\|u_{\mathcal{P}}(S, G) - m_{\mathcal{P}}\|^2$, and define $A_{\mathcal{P}'}$ analogously. With all terms finite,*

$$R_{\mathrm{ref}}(u_{\mathcal{P}}; \mathcal{P}) - R_{\mathrm{ref}}(u_{\mathcal{P}'}; \mathcal{P}') = \Delta(\mathcal{P}, \mathcal{P}') + A_{\mathcal{P}} - A_{\mathcal{P}'}. \tag{A.5}$$

For a model-count penalty λK, $\lambda \geq 0$, the refined predictor has strictly lower penalized reference risk exactly when $\Delta(\mathcal{P}, \mathcal{P}') > A_{\mathcal{P}'} - A_{\mathcal{P}} + \lambda(K' - K)$.

Proof of Corollary A.1.

Proof. Apply (A.1) with input (S, G) and then with (S, G'). Subtract the resulting expressions $B(\mathcal{P}) + A_{\mathcal{P}}$ and $B(\mathcal{P}') + A_{\mathcal{P}'}$, and add the difference in penalties. □

The nonnegative residual $A_{\mathcal{P}}$ includes approximation, estimation, and optimization effects; no particular scaling with K or sample count is assumed. The penalty gives a decision criterion for the reference problem, not a proof of optimality for the validation-increment switching rule.

Stage labels can recover predictive distinctions obscured by a compressed context, but generation and switching introduce no new future observations. Appendix A.6 uses this information constraint to quantify a horizon-dependent error floor under persistent innovations.

A.5 From reference accuracy to actual rollout accuracy

A candidate changes its later evaluation states (Ross et al., 2011). Couple a candidate and frozen reference chain through the same initial history and targets. Write $f_j = f_{\theta_{\kappa_{\mathcal{P}}(j)}}$, with reference predictors and states $\bar{f}_j, \bar{S}_j$.

Theorem A.2 (Propagation and reference-to-rollout risk bound)**.** *Assume each f_j is L_j-Lipschitz between the candidate and reference states under consideration, and all variables below have finite second moments. Define*

$$e_j = \left(\mathbb{E}\|S_j - \bar{S}_j\|^2\right)^{1/2}, \quad \rho_j = \left(\mathbb{E}\|f_j(\bar{S}_{j-1}) - \bar{f}_j(\bar{S}_{j-1})\|^2\right)^{1/2}, \quad a_j = \sqrt{1 + L_j^2}. \tag{A.6}$$

With $e_0 = 0$ and empty products equal to one,

$$e_j \leq a_j e_{j-1} + \rho_j \leq \sum_{i=1}^{j} \rho_i \prod_{r=i+1}^{j} a_r. \tag{A.7}$$

Let $R_{\mathrm{ref}}(F) = \sum_j w_j \mathbb{E}\|f_j(\bar{S}_{j-1}) - Y_j\|^2$ and $D_F = (\sum_j w_j L_j^2 e_{j-1}^2)^{1/2}$. Then

$$\left|\sqrt{R_{\mathrm{roll}}(F)} - \sqrt{R_{\mathrm{ref}}(F)}\right| \leq D_F. \tag{A.8}$$

Proof of Theorem A.2.

Proof. Because $h \le L$, the retained context and appended output occupy disjoint coordinates. Let A be the projection onto the last $L-h$ rows of a state. The squared distance between two candidate updates is

$$\|A(s-\bar{s})\|^2 + \|f_j(s) - f_j(\bar{s})\|^2 \le (1+L_j^2)\|s-\bar{s}\|^2.$$

At the same reference state, the distance between candidate and reference updates is exactly $\|f_j(\bar{s}) - \bar{f}_j(\bar{s})\|$. Insert the candidate update at $\bar{S}_{j-1}$ and apply the triangle inequality in L^2 to obtain the first inequality in (A.7). Induction starting from $e_0 = 0$ gives the sum of products. Finally, put the errors of all forecast blocks in the weighted Hilbert space with norm $(\sum_j w_j \mathbb{E}\|Z_j\|^2)^{1/2}$. The reverse triangle inequality bounds the difference of the two risk square roots by

$$\left(\sum_j w_j \mathbb{E}\|f_j(S_{j-1}) - f_j(\bar{S}_{j-1})\|^2\right)^{1/2} \le D_F,$$

which proves (A.8). □

For two candidates F_1, F_2 on this common reference law, a sufficient condition for $R_{\mathrm{roll}}(F_1) < R_{\mathrm{roll}}(F_2)$ is

$$\sqrt{R_{\mathrm{ref}}(F_1)} + D_{F_1} < \sqrt{R_{\mathrm{ref}}(F_2)} - D_{F_2}. \tag{A.9}$$

Indeed, the left side upper-bounds $\sqrt{R_{\mathrm{roll}}(F_1)}$ and the right side lower-bounds $\sqrt{R_{\mathrm{roll}}(F_2)}$. Thus a reference improvement must dominate the possible state-distribution change. Here ρ_j measures disagreement with the reference predictor, not error against the true future. The factors a_j are conservative upper bounds, not a claim that realized errors grow monotonically or that the models are contractive.

Example A.1 (Perfect reference predictions can worsen a rollout)**.** Let $L = h = d = 1$, $N = 2$, $w_1 = w_2 = 1/2$, $S_0 = 0$, and deterministic targets $(Y_1, Y_2) = (1, 0)$. A reference predictor $\bar{f}_1 = \bar{f}_2 = 0$ generates only zero states and has risk $1/2$. For $M > 1$, take $f_1(s) = 1$ and $f_2(s) = Ms$. On the reference states, the candidate predicts $(1, 0)$ and has $R_{\mathrm{ref}}(F) = 0$. Its own rollout predicts $(1, M)$, however, so $R_{\mathrm{roll}}(F) = M^2/2 > 1/2$. Here $e_1 = 1$ and $D_F = M/\sqrt{2}$, attaining (A.8).

This subsection compares reference and rollout risks at a common horizon. We next use the true trajectory as the reference to study how error bounds depend on the number of rollout calls.

A.6 An exponential upper bound for reuse and a linear lower bound for switching

To distinguish propagation from unavoidable forecast uncertainty, define the true state $S_j^\star = T(S_{j-1}^\star, Y_j)$ with $S_0^\star = S_0$, and use the error measures defined in equation (7) in the main text. Here M_j is the expected squared error of block j, not the accumulated loss of the first j blocks; coordinate MSE is $M_j/(hd)$. Since the last h rows of the two states are $\widehat{Y}_j$ and Y_j, respectively, $\sqrt{M_j(F)} \le q_j(F)$. The following upper and lower bounds have separate sufficient conditions and concern population errors conditional on fitted parameters and the selected partition.

Proof of Theorem 3.2.

Proof. Insert $\Phi(S_{j-1}^\star)$ into $S_j - S_j^\star$. At the same true state, $\Phi(S_{j-1}^\star)$ and $T(S_{j-1}^\star, Y_j)$ differ only in their appended blocks. The triangle inequality in L^2 therefore gives

$$q_j(F) \le \gamma q_{j-1}(F) + \varepsilon.$$

Starting from $q_0(F) = 0$, expansion yields $q_j(F) \le \varepsilon \sum_{i=0}^{j-1} \gamma^i$. Summing the geometric series and using $M_j(F) \le q_j(F)^2$ proves (8). □

This is a discrete Gronwall estimate, the standard mechanism in global error analysis of one-step integration (Hairer et al., 1993). For a forward Euler update with step size η, a vector field with Lipschitz constant $\ell > 0$, and local truncation error at most $C\eta^2$, one may take $\gamma = 1 + \eta\ell$ to obtain $q_j \le (C\eta/\ell)(e^{\ell j\eta} - 1)$. The analogy concerns error propagation; it does not assume that a forecasting network is a consistent ODE discretization. Here γ bounds the full state update and must not be identified with the output Lipschitz constant L_j in Appendix A.5.

The bound permits exponential growth without requiring every fitted model to exhibit it. It is attained in one dimension: take $L = h = d = 1$, $S_0 = 0$, zero true targets, and $f(s) = \gamma s + \varepsilon$. Then $\widehat{Y}_j = \varepsilon(\gamma^j - 1)/(\gamma - 1)$. Thus these local-error and Lipschitz conditions alone do not permit a uniformly slower growth order.

Uncertainty that survives switching. Switching supplies no new observations. A strictly growing error floor additionally requires new disturbances in the true process to have persistent effects on distant targets. The following sufficient condition generalizes the growth of random-walk forecast variance with horizon (Hyndman & Athanasopoulos, 2021).

The definition is stated in Definition 3.3.

Zero conditional mean makes each innovation unpredictable from the preceding true history; independence and Gaussianity are unnecessary. The coefficient condition prevents past innovations from fading away over the analyzed horizon. Nonzero one-step noise variance alone is insufficient.

Averaged lower bounds. Consequently,

$$R_{\mathrm{roll}}(F) \ge c_*^2\sigma_*^2 \sum_{j=1}^{N} w_j j, \qquad \frac{1}{N}\sum_{j=1}^{N} M_j(F) \ge c_*^2\sigma_*^2 \frac{N+1}{2}. \tag{A.10}$$

Proof of Theorem 3.3.

Proof. Condition on fitted parameters and the partition. Induction in (1) makes every generated state and output $\mathcal{G}_0$-measurable, including across handovers. Write $\widehat{Z}_j = \langle v, \widehat{Y}_j\rangle$. The tower property gives $\mathbb{E}[\xi_i \mid \mathcal{G}_0] = 0$. For $i < r$, ξ_i is $\mathcal{G}_{r-1}$-measurable, so $\mathbb{E}[\xi_i\xi_r] = \mathbb{E}[\xi_i\mathbb{E}(\xi_r \mid \mathcal{G}_{r-1})] = 0$. Hence

$$\mathbb{E}(\widehat{Z}_j - Z_j)^2 = \mathbb{E}(\widehat{Z}_j - \mu_j(S_0))^2 + \sum_{i=1}^{j} c_{j,i}^2\mathbb{E}[\xi_i^2]. \tag{A.11}$$

The cross terms vanish by zero conditional mean given initial information and orthogonality between innovations; square integrability ensures that all products are integrable. Cauchy–Schwarz and $\|v\| = 1$ imply $M_j(F) \ge \mathbb{E}(\widehat{Z}_j - Z_j)^2$. Dropping the nonnegative fitting term and applying (10) proves (11). Summing with weights w_j and $1/N$, respectively, proves (A.10). □

The lower bound is attained by a random walk: take $L = h = d = 1$ and $Y_j = S_0 + \sum_{i=1}^{j}\xi_i$, with innovation variance σ_*^2. Conditional-mean predictions $\widehat{Y}_j = S_0$ in every stage give $M_j = j\sigma_*^2$. In a mean-reverting process, by contrast, innovation effects may decay and distant-horizon errors need not keep growing. Definition 3.3 is thus a substantive process assumption, not a consequence of stagewise training.

Connection to error curves. These results give an exponential worst-case upper bound for model reuse and a linear error floor that persists under switching. The lower bound concerns MSE; for coordinate RMSE it gives only $c_*\sigma_*\sqrt{j/(hd)}$. It also applies to a single model, so it neither orders the two methods nor establishes exactly linear switching error. Experiments should assess growth and innovation persistence on a common per-block MSE scale. A switching curve close to the linear floor would support the interpretation that additional propagation error is small, but this interpretation requires empirical validation.

B PUBLIC-BOUNDARY CONSTRUCTION: DETAILS AND PROOFS

Equal-length partitions assign the same reuse duration to every model, although generated intervals may require different degrees of adaptation. We seek to retain a model while its error grows moderately and introduce a successor when accuracy deteriorates or growth accelerates, approaching dense-stage accuracy with fewer parameter sets. We study two alternative criteria: a **linear error threshold** and a **slope-acceleration threshold**. The segmentation decision relates to the statistics, search procedures, and segment-count constraints studied in change-point detection (Truong et al., 2020). Here the segments specify predictor reuse, and handovers need not identify changes in the true dynamics.

B.1 OFFLINE CONSTRUCTION AND TRAINING INTERVALS

For a common horizon N, separate parameter-training data $\mathcal{D}_{\mathrm{tr}}$, boundary-construction data $\mathcal{D}_{\mathrm{b}}$, chain-selection data $\mathcal{D}_{\mathrm{s}}$, and a final test set. Distinct validation roles prevent repeated boundary search from being mistaken for independent performance evidence (Cawley & Talbot, 2010). Normalization uses training data only and is shared across stages. Boundaries are fixed before deployment; test-time switching uses position, without future losses or online training.

Suppose calls through $a = b_{k-1}$ have been accepted. Freeze the preceding models and generate training and construction entrances S_a using initial observations and accepted predictions. Since b_k is unknown, prescribe a training-unroll length ℓ_{fit} and use $\widetilde{I}_k = \{a+1, \dots, \min(a+\ell_{\mathrm{fit}}, N)\}$ in place of I_k in (4), renormalizing its weights. Train through this interval, then freeze parameters for boundary probing. The accepted reuse interval I_k may be longer or shorter than $\widetilde{I}_k$. Reuse beyond the training unroll is validation-monitored extrapolation; it is not joint training over the entire accepted stage. Both criteria use the same unroll length, initialization protocol, and optimization budget per model.

Reuse f_{θ_k} from each generated construction entrance. Let $\widehat{Y}_{k,r}^{(i)}$ be its local output r, at global position $j = a + r$. Monitor per-block coordinate MSE: See equation (16) in the main text. True futures score predictions but never update generated states. This statistic is neither cumulative prefix error nor renormalized after handovers. Both policies warm up for $r_0 = m + 1$ calls, where $m \geq 1$, with $\ell_{\mathrm{fit}} \geq r_0$; a shorter remaining interval is completed as a terminal stage.

B.2 TWO SWITCHING SIGNALS

Policy 1: a linear error threshold. For $\alpha_0, \beta_0 \geq 0$ chosen during construction, define, at $r > r_0$,

$$\tau(j) = \alpha_0 + \beta_0 j, \qquad H_{k,r}^{\mathrm{lin}} = \mathbf{1}\{\widehat{E}_{k,r} > \tau(a+r)\}. \tag{B.1}$$

The intercept allows initial error and the slope specifies acceptable horizon-dependent growth. Global position $a + r$ prevents handovers from resetting accumulated uncertainty. The coordinate MSE floor in Theorem 3.3 motivates a linear scale, but is not an upper bound and does not determine α_0, β_0. If its assumptions hold uniformly over unbounded horizons and $\beta_0 < c_*^2 \sigma_*^2/(hd)$, the envelope with any finite intercept eventually lies below the population MSE floor. Population error then cannot remain below the envelope; a corresponding finite-validation alarm guarantee would additionally require control of estimation error. Thresholds must therefore be calibrated on construction data and their accuracy cost assessed by independent chain selection.

Policy 2: slope acceleration relative to a recent baseline. Let $g_{k,r} = \widehat{E}_{k,r} - \widehat{E}_{k,r-1}$ for $r \geq 2$. With equally spaced calls, this is the discrete slope per call. Define

$$\begin{aligned} B_{k,r} &= \mathrm{median}(g_{k,r-m}, \dots, g_{k,r-1}), \\ A_{k,r} &= g_{k,r} - \lambda \max\{B_{k,r}, s_{\min}\}, \qquad H_{k,r}^{\mathrm{slope}} = \mathbf{1}\{A_{k,r} > \zeta\}, \end{aligned} \tag{B.2}$$

where $\lambda > 1$, $s_{\min} > 0$, and $\zeta \geq 0$. For even windows, the median averages the middle two entries. Compare the current increment with the preceding m increments before inserting it into the window. The positive floor handles zero or negative baselines, while ζ adds absolute slack. Each stage builds its own window; a change of predictor is not included in a within-model slope.

The signal detects a further increase in the growth rate, connecting to the amplification mechanism in Appendix A.6. It is not a test that identifies an exponential function: quadratic curves, jumps, or noise can also trigger it, whereas high but flat error may not. The two policies address error level and local relative growth, respectively, and should be compared on a common loss scale and model budget.

B.3 Confirmation, handover, and termination

After choosing one signal $H_{k,r}$, increment a counter when it equals one and reset the counter otherwise; do not count during warm-up. On $p \geq 1$ consecutive alarms at probe r, end the old stage at

$$b_k = a + r - 1. \tag{B.3}$$

Retain all earlier probe blocks, discard the current block, and restore the state immediately before that call. The successor predicts again from $b_k + 1$ without the discarded context. For $p = 2$, the first flagged block is retained and the second discarded. This rollback is offline only. Confirmation rejects isolated alarms, but $p > 1$ may retain threshold violations, so the accepted interval is not claimed to satisfy a pointwise error envelope. Generate the next training entrance by rolling the final frozen model to b_k, rather than retaining states produced by changing parameters during optimization.

Construction procedure. Given N, a model budget $K_{\max}$, and prespecified criterion parameters:

1. Set $a = 0$, $k = 1$, and cache training and construction initial states.
2. Train on $\widetilde{I}_k$ and freeze the model. Probe from the construction entrance, updating the chosen signal and consecutive counter after warm-up.
3. On confirmation with $k < K_{\max}$, accept (B.3), restore the construction state, and generate the training handover state. Set $a = b_k$, increment k, and repeat Step 2.
4. On reaching N, set $b_k = N$ and finish. Once the model budget is exhausted, complete the remaining interval with the last model and record unhandled alarms; this tail is not thereby certified to satisfy the threshold.

A stage entered with fewer remaining calls than the warm-up length is trained for that tail and completed without further splitting. This procedure fixes the forecasting task length. Variable-coverage studies require a separate stopping criterion and accuracy comparisons over a common horizon.

With cached handover states and one discarded probe per switch, construction uses $N + K - 1$ prediction calls per validation case; parameter training and training-context generation are additional costs. Deployment still uses N calls per case, with K parameter sets and $K - 1$ handovers. Fewer models may reduce weight storage, training, and loading costs, but savings also depend on construction search; a smaller K does not establish a reduction in total computation.

B.4 Criterion properties and selection of fewer models

Proposition B.1 (Slope response and perturbation margins)**.** *Within a fixed stage, let E_r be population coordinate MSE under continued reuse, and let $A_r(E)$ replace $\widehat{E}$ with E in* (B.2)*. If $|\widehat{E}_t - E_t| \leq \epsilon$ for $t = r - m - 1, \ldots, r$, then*

$$|A_r(\widehat{E}) - A_r(E)| \leq 2(1 + \lambda)\epsilon. \tag{B.4}$$

For an affine curve $E_r = A + br$ with $b \geq 0$, $A_r(E) \leq 0$, so no slope alarm occurs when $\zeta \geq 2(1 + \lambda)\epsilon$. For $E_r = A + C\rho^r$ with $C > 0, \rho > 1$, if the median is at least $s_{\min}$, write $d_m(\rho) = \mathrm{median}(\rho^{-1}, \ldots, \rho^{-m})$. Then

$$A_r(E) = C(\rho - 1)\rho^{r-1}[1 - \lambda d_m(\rho)]. \tag{B.5}$$

Suppose the median-floor condition holds at each of p consecutive monitored calls, and the same estimation-error bound holds on the union of their windows. If the bracket is positive and this expression exceeds $\zeta + 2(1 + \lambda)\epsilon$ at each call, switching is confirmed by the last of them unless the horizon or model budget has been exhausted.

Proof of Proposition B.1.

Proof. Each increment changes by at most 2ϵ. Coordinatewise perturbations of this size change every order statistic, and hence the median, by at most 2ϵ. Since $x \mapsto \max(x, s_{\min})$ is 1-Lipschitz, the triangle inequality proves (B.4). Affine increments all equal b, giving $A_r(E) = b - \lambda \max(b, s_{\min}) \le 0$. For the exponential curve, the current increment is $C(\rho-1)\rho^{r-1}$ and the increment i calls earlier is ρ^{-i} times as large. Taking their median proves (B.5). Applying the perturbation bound at every one of the p calls puts each observed score above ζ, so the consecutive counter confirms the switch. □

For $m = 4$, $d_m(\rho) = (\rho^{-2}+\rho^{-3})/2$, so ideal slope detection requires $2\rho^3/(1+\rho) > \lambda$. This exposes a limitation of a moving baseline: it can track slowly growing exponential errors without triggering when this condition fails. An unbounded exponential curve eventually exceeds any fixed linear envelope. These results characterize detection, without treating the upper bound in Theorem 3.2 as an equation for observed errors.

Selecting few models within a dense-baseline tolerance. A local alarm does not guarantee that its successor improves the forecast. We assess model savings against a dense-stage baseline under the same deployment budget.

Construct complete candidates $F_1, \dots, F_Q$ and baseline F_0 using training and construction data, then freeze them. Define $\ell(F, z) = \sum_j w_j \|\widehat{Y}_j - Y_j\|^2/(hd)$ and $\mathcal{R}(F) = \mathbb{E}\ell(F, z)$. On n_{s} independent selection cases, compute

$$\widehat{D}_q = \frac{1}{n_{\mathrm{s}}} \sum_{i=1}^{n_{\mathrm{s}}} [\ell(F_q, z_i) - \ell(F_0, z_i)], \qquad r_n = B\sqrt{\frac{2\log(2Q/\delta)}{n_{\mathrm{s}}}}. \tag{B.6}$$

Here $B > 0$ is an assumed-valid uniform loss bound, $\delta \in (0, 1)$, and $\Delta_{\mathrm{tol}} \ge 0$ is a prespecified risk tolerance. Resolve ties by fixed ordering and select

$$\mathcal{C} = \{0\} \cup \{q : 1 \le q \le Q,\ \widehat{D}_q + r_n \le \Delta_{\mathrm{tol}}\}, \qquad q^* \in \arg\min_{q \in \mathcal{C}} K(F_q), \tag{B.7}$$

Theorem B.1 (Accuracy tolerance and model count in a finite candidate set)**.** *Conditional on construction data, assume that selection cases are iid and independent of all candidate chains, and each chain's per-case loss lies in* $[0, B]$ *almost surely. Set* $D_q = \mathcal{R}(F_q) - \mathcal{R}(F_0)$*. With probability at least* $1 - \delta$*, the rule* (B.7) *satisfies*

$$\begin{aligned} &\mathcal{R}(F_{q^*}) \le \mathcal{R}(F_0) + \Delta_{\mathrm{tol}}, \\ &K(F_{q^*}) \le \min\big(\{K(F_0)\} \cup \{K(F_q) : D_q \le \Delta_{\mathrm{tol}} - 2r_n\}\big). \end{aligned} \tag{B.8}$$

Proof of Theorem B.1.

Proof. Each paired loss difference lies in $[-B, B]$. Hoeffding's inequality (Hoeffding, 1963) gives $\Pr(|\widehat{D}_q - D_q| > r_n) \le 2\exp[-n_{\mathrm{s}} r_n^2/(2B^2)] = \delta/Q$. A union bound over Q candidates gives simultaneous deviations at most r_n with probability at least $1 - \delta$. On this event, an accepted non-baseline candidate has $D_q \le \widehat{D}_q + r_n \le \Delta_{\mathrm{tol}}$; the baseline has zero gap. Conversely, $D_q \le \Delta_{\mathrm{tol}} - 2r_n$ implies $\widehat{D}_q + r_n \le \Delta_{\mathrm{tol}}$, making every such candidate eligible. Minimizing model count over the eligible set proves both claims. □

Thus a candidate with fewer models and sufficient accuracy margin yields both tolerance control and model savings under these conditions. The theorem does not ensure that such a candidate exists or that greedy boundaries globally minimize model count over all partitions. Bounded loss and independent selection cases are additional assumptions: ordinary MSE and overlapping windows from one long series do not automatically satisfy them. Do not retrain or revise boundaries on $\mathcal{D}_{\mathrm{s}}$ after selection; report full-rollout accuracy and model count on the final test set. Without a valid error radius, a validation tolerance remains an empirical selection criterion rather than a population guarantee.

C Sample-Adaptive Training and Routing Diagnostics

This appendix separates slope-based sample-specific training, inference on a frozen chain, and risk-prediction diagnostics. Individual stage models remain shared across samples. The true-error policies deliberately use privileged target information; they are not deployable forecasters.

C.1 Training intervals aligned with individual handovers

Let $a_k^{(i)}$ count accepted blocks when case i enters stage k, with $a_1^{(i)} = 0$ and active set $\mathcal{A}_k = \{i : a_k^{(i)} < N\}$. Using the training-unroll length ℓ_{fit}, define

$$\begin{aligned} \widetilde{I}_k^{(i)} &= \{a_k^{(i)} + 1, \dots, \min(a_k^{(i)} + \ell_{\mathrm{fit}}, N)\}, \\ W_k^{(i)} &= \sum_{j \in \widetilde{I}_k^{(i)}} w_j, \qquad U^{(i)}_{a_k^{(i)}}(\theta) = S^{(i)}_{a_k^{(i)}}. \end{aligned} \tag{C.1}$$

Generate entrances with the frozen accepted prefix and use (18). Gradients pass through the current model's unroll; predecessor parameters stay fixed. Normalization by $W_k^{(i)}$ accommodates shortened terminal windows.

After fitting a stage, continue each training case and apply the slope rule to its realized block error,

$$e_{k,r}^{(i)} = \frac{1}{hd} \| \widehat{Y}_{k,r}^{(i)} - Y^{(i)}_{a_k^{(i)}+r} \|^2. \tag{C.2}$$

Warm-up, the recent-increment window, and consecutive confirmation follow the training protocol. A confirmed alarm at probe r rejects that probe and gives $a_{k+1}^{(i)} = a_k^{(i)} + r - 1$. Each successor learns the window following that individual entrance. Cases reaching N exit the active set; construction stops at the budget or when none remain. Targets select training boundaries and supervise fitting, but never replace generated values.

After construction, all forecasters are frozen. The inference-policy comparison uses identical forecaster checkpoints and training entrances, even when the inference policy uses a linear threshold or adjacent-model comparison. It does not compare independently retrained chains, and the reported comparison does not include a learned-policy forecaster replay or refit.

C.2 Risk supervision, candidate coverage, and generalization

Definition 3.4 specifies the original MSE query. Available inputs include initial and generated histories, candidate predictions, and global/local positions. Future targets, realized losses, and true-error handover positions cannot be deployment inputs. Separate risk-training sequences supply labels for frozen forecasters; disjoint calibration/selection data determine learned-policy settings. Oracle test targets are used only for the explicitly identified diagnostics, never for fitting forecasters or risk predictors.

For linear and slope policies, supervise the error of the active model. A paired comparison additionally needs coverage of the immediate successor on the *current* generated context, including contexts on which it was not selected. Training only on accepted low-error outputs censors failures; training only on a model's original entrance distribution may miss comparison queries. Collecting diagnostic probes on training or validation trajectories can address coverage without refitting the forecasters. Deployment-induced distribution mismatch is related to the issue studied by DAgger (Ross et al., 2011), but no interactive-learning guarantee transfers automatically to this offline experiment.

The observed gap between PredCmp and TrueCmp establishes that the tested learned policy does not recover future-informed routing performance. It does not identify generalization as the sole cause. Relevant mechanisms include loss heterogeneity, errors in ranking candidates near the switching margin, sparse late-stage coverage, and route-induced distribution changes. Separate held-out prediction, calibration, and paired-ranking diagnostics would distinguish these mechanisms.

A conditional mean is not a realized future error. Here the comparison loss ℓ is per-sample sRMSE; the paired plug-in policy requires estimates of its conditional mean for each candidate. For estimated losses with individual errors at most ϵ, the difference score $d = \ell_{k+1} - (1 - \rho)\ell_k$ changes by at

most $(2-\rho)\epsilon$, by the triangle inequality. Agreement with a realized-error oracle would require control relative to the *realized* losses, a stronger condition than mean-risk calibration. Moreover, small average error does not ensure correct ordering near $d = 0$.

C.3 Point and distributional error predictors

Point prediction. For the coordinate-MSE interface, a nonnegative MLP can minimize

$$\widehat{\phi}_k \in \arg\min_{\phi} \mathbb{E}_{(Z,e)\sim\widehat{\nu}_k}(q_\phi(Z) - e)^2, \tag{C.3}$$

where $\widehat{\nu}_k$ is the empirical query distribution. With finite second moment of e, the population target is $\mathbb{E}[e \mid Z]$ (Gneiting, 2011). Quantile or distributional heads are alternatives (Tagasovska & Lopez-Paz, 2019), but changing the target requires recalibrating the decision policy. A direct advantage predictor is another possible interface; it should be reported as a distinct implementation rather than silently equated with two calibrated error predictors.

Distributional prediction as an architectural alternative. Conditional diffusion can model $R = \mathrm{vec}(\widehat{Y} - Y)/\sqrt{hd}$ given Z, so $e = \|R\|^2$. CARD (Han et al., 2022) and TimeGrad (Rasul et al., 2021) motivate this modeling option, not superior routing in the present experiments. A conditional denoising objective is (Ho et al., 2020)

$$\begin{aligned} R_t &= \sqrt{\bar{\alpha}_t}R + \sqrt{1-\bar{\alpha}_t}\,\xi, \qquad \xi \sim \mathcal{N}(0, I), \\ \mathcal{L}_{\mathrm{diff},k}(\phi) &= \mathbb{E}_{R,Z,t,\xi}\|\xi - \epsilon_\phi(R_t, t; Z)\|^2. \end{aligned} \tag{C.4}$$

Here t indexes diffusion noise rather than forecast lead. Conditional residual samples estimate mean MSE as

$$q_{\phi_k}(Z) = \frac{1}{B_{\mathrm{MC}}}\sum_{b=1}^{B_{\mathrm{MC}}} \|R^{(b)}\|^2. \tag{C.5}$$

Squaring the mean residual would omit conditional variance. For the sRMSE comparison, directly supervise sRMSE or compute sRMSE for each sampled residual on the comparison interval before averaging over samples. The square root of predicted mean scaled MSE is generally not conditional mean sRMSE. Learning multimodal residuals may help represent uncertainty, but does not by itself solve ranking, calibration, or distribution shift. Any evaluation of this alternative would need to account for sampling variability and denoising cost. This discussion does not assert that a diffusion implementation has been evaluated or that it outperforms the tested predictor.

C.4 Frozen-chain inference and access to target information

All policies begin with the same initial observations and ordered forecaster chain. They preserve accepted generated history, move forward through adjacent stages, and use the final model when the chain is exhausted. Changing routing is allowed to change subsequent generated contexts; it does not change any forecaster weights.

Absolute error rules. PredLin/PredSlope use estimated errors in (12); TrueLin/TrueSlope substitute realized errors. The linear rule uses global time, and the slope window resets on a handover. Warm-up, persistence, and terminal handling are matched within each predicted/true pair. The latter pair reads future targets to make decisions and is therefore diagnostic. Retaining generated state does not remove that information advantage.

Adjacent-model comparison. TrueCmp reads both candidates' realized sRMSE values and applies (19); PredCmp uses a learned replacement. The comparison should use identical inputs and target support for both candidates. The formal rule retains a zero-error current predictor and requires a genuine relative improvement. With current-block acceptance, accept the selected candidate and discard the other, then advance the global index; do not read the next unseen block to decide the present one. The comparison uses per-sample sRMSE, computed as in Appendix D.1 but restricted to the common comparison interval, using the same channel scales for both candidates. A five-percent relative reduction means $\ell_{k+1,j} \le 0.95\ell_{k,j}$. For squared sRMSE, the equivalent factor is

$0.95^2 = 0.9025$, a 9.75% reduction; applying a 5% threshold to MSE would define a different rule. This local decision is distinct from evaluation at forecast lead 500.

A locally smaller realized loss does not imply the lowest complete-trajectory loss: the selected output changes future contexts, and a forward-only switch consumes a stage opportunity. Thus TRUECMP is a greedy privileged comparator, not a proven upper bound on achievable accuracy or a lower bound on full-horizon loss.

Cost accounting. For absolute rules, one rejected current probe per handover gives $N + K_i - 1$ forecaster calls under the stated acceptance convention. For adjacent comparison, probing a successor costs a call even when no switch occurs. With one current and at most one successor call per block and candidate caching, cost is $N + C_i$, where C_i counts successor probes. Predictor evaluations, batching, and memory add further implementation-dependent costs. Oracle target access is reported as privileged information, not assigned a fictitious deployment latency.

C.5 CONDITIONAL ROUTING GUARANTEES AND OBSERVED FAILURES

Proposition 3.1 concerns learned versus ideal conditional-risk routing for the linear and slope policies. Its conclusion is preserved below; it is not a theorem about TRUECMP or the empirical superiority of sample-specific adaptation.

Proof of Proposition 3.1.

Proof. Couple both procedures on the same sequence and, where needed, the same internal random draws. Before their first differing decision they have identical accepted states, model indices, ages, and tentative forecasts, hence identical query features. The linear margin preserves the alarm sign. For the slope rule, the perturbation bound in Proposition B.1 is algebraic and applies to the conditional-risk sequence, preserving its alarm sign as well. Warm-up decisions are fixed. Therefore alarm histories, confirmation counters, and handovers agree by induction, including the treatment of the horizon and budget. On $\mathcal{G}$ the complete forecasts, and hence losses, coincide. For each x, their loss-indicator difference is bounded in absolute value by $\mathbf{1}_{\mathcal{G}^c}$, proving the CDF bound. The inequalities $F_{\Pi_\mu} - \delta \le F_{\Pi_q} \le F_{\Pi_\mu} + \delta$ then give the generalized-inverse bounds in (15). □

The assumptions require simultaneous accuracy and separation from thresholds along the route. Average calibration alone is insufficient, and a reliable approximation to a poor rule can remain a poor forecaster. Conditional-risk routing can also differ from routing by realized losses because unpredictable innovations remain.

In this proposition Q_η acts on full-horizon squared sample losses. The reported $q90$ and $q95$ instead use per-sample sRMSE at forecast lead 500 (Appendix D.1). Agreement of complete forecasts also permits the same coupling argument for endpoint loss, but neither the growth bounds nor quantile transfer implies an improvement in these metrics. The measured PredCmp–TrueCmp differences are empirical comparisons on common cases and frozen checkpoints.

D EXPERIMENTAL DETAILS AND SUPPLEMENTARY RESULTS

This appendix reports data sources, endpoint evaluation, and supplementary comparisons of generated-context training, stage counts, and adaptive model use.

D.1 DATASETS AND EVALUATION PROTOCOL

Datasets. MAST contains tokamak physical-quantity sequences curated by TokaMark (Rousseau et al., 2026) from FAIR-MAST (Jackson et al., 2024). Extended TEP provides chemical-process simulations covering healthy and faulty conditions and changes in operation (Reinartz et al., 2021), with data in the DTU repository. C-MAPSS contains simulated turbofan degradation trajectories (Saxena & Goebel, 2008). We forecast future physical variables on all three datasets; the TEP and C-MAPSS evaluations are sequence forecasting, rather than fault classification or remaining-useful-life prediction.

Information available during rollout. Deployable forecasts use initial observations and generated histories without refreshing future target observations. All backbones within a comparison use the same available information. TrueCmp is a privileged diagnostic: it preserves generated histories but uses future test targets to select between adjacent forecasters. This access is excluded from deployable policies.

Endpoint metrics. For test case i with forecast origin t_i, let $s_c > 0$ denote the fixed training-data scale for channel c. At forecast lead 500, define

$$r_i(500) = \left[\frac{1}{d} \sum_{c=1}^{d} \left(\frac{\widehat{X}^{(i)}_{t_i+500,c} - X^{(i)}_{t_i+500,c}}{s_c} \right)^2 \right]^{1/2},$$

$$\bar{r}_{500} = \frac{1}{n} \sum_{i=1}^{n} r_i(500),$$

$$q90 = \widehat{Q}_{0.9}(r_1(500), \ldots, r_n(500)),$$

$$q95 = \widehat{Q}_{0.95}(r_1(500), \ldots, r_n(500)).$$

Here n is the number of test cases, d the number of forecast channels, and $\widehat{Q}$ an empirical quantile. Tables 1 and 2 use this endpoint metric; $\bar{r}$ abbreviates $\bar{r}_{500}$ in adaptive-result tables. The mean averages per-case roots, not squared errors before rooting. The quantiles measure variation across cases at the endpoint, not across forecast times. Thus terminal-time error and high-quantile error describe different aspects of performance. Only point estimates are reported; no statistical significance or full-horizon dominance is claimed.

The theoretical growth bounds and absolute error signals use block MSE. Adjacent-model comparison uses per-sample sRMSE on the common decision interval and switches at a relative reduction of at least 5%. The local decision statistic and the endpoint evaluation have distinct roles; an MSE lower bound is not a linear sRMSE bound.

Comparisons. The context grid contains 54 error–time pairs: three datasets, three backbones, $K = 5, 10, 20$, and observed/generated contexts. In `pred`/G, successors train on histories generated by frozen predecessors; `real`/R uses observed histories. Both use true futures as supervision. The MAST setup uses 200 historical points and predicts 500 future points in blocks of 10. TFT (Lim et al., 2021), DAG (Qiu et al., 2026), and Timer-XL (Liu et al., 2025) provide the forecasting backbones.

All fixed-stage and adaptive forecasters in Table 2 use generated-context training. The table contains 45 fixed/public configurations across the three datasets and six MAST sample-routing configurations. PredCmp and TrueCmp use the same frozen slope-trained forecasters and differ in their inference-time error information. Sample-routing results are limited to MAST. TrueCmp is an exploratory diagnostic, not evidence of deployable routing reliability.

Cost and model use. Training durations include fixed-stage forecaster fitting and context generation, expressed as hours:minutes. Adaptive model use is $\bar{K} = n^{-1} \sum_i K_i$, where K_i counts distinct forecasters contributing accepted predictions over the rollout; rejected probes are excluded. This count does not measure inference latency or total model-bank storage. No adaptive runtime speedup is inferred from model counts. Table D.1 extracts the $K = 10$ context comparison, and Table D.2 lists its MAST subset.

D.2 Effect of generated-context training

Across all 27 matched configurations, `pred` has lower endpoint mean sRMSE than `real`. Relative reductions, $(E_{\text{real}} - E_{\text{pred}})/E_{\text{real}}$, range from 17.63% (C-MAPSS, Timer-XL, $K = 5$) to 96.24% (C-MAPSS, DAG, $K = 20$). The direction holds within each of the three datasets and backbones, consistent with generated-context adaptation (Appendix A.2). This is a descriptive comparison of reported means, without a statistical significance claim.

Table D.1: Context comparison at $K = 10$. Errors are mean per-sample sRMSE at step 500; training durations include context generation (hours:minutes). Bold marks the smaller paired error.

Dataset	Backbone	$\bar{r}^{\mathrm{R}}_{500}$	$\bar{r}^{\mathrm{G}}_{500}$	T_{R}	T_{G}
MAST	TFT	0.05815	**0.03726**	3:59	8:37
MAST	DAG	0.77653	**0.07074**	2:53	6:04
MAST	Timer-XL	0.05035	**0.03596**	2:05	3:39
TEP	TFT	0.24065	**0.11991**	5:11	11:57
TEP	DAG	1.02337	**0.26029**	3:16	6:38
TEP	Timer-XL	0.14425	**0.09546**	2:11	3:25
C-MAPSS	TFT	0.01772	**0.00819**	6:55	14:32
C-MAPSS	DAG	0.16413	**0.01279**	5:07	10:56
C-MAPSS	Timer-XL	0.00512	**0.00229**	4:28	8:52

Table D.2: Fixed-stage results on MAST. Each K reports step-500 mean sRMSE and training time including context generation (hours:minutes). Bold indicates the lower error between the two training inputs for the same backbone and stage count.

		$K = 5$		$K = 10$		$K = 20$	
Backbone	Input	$\bar{r}_{500}$	Time	$\bar{r}_{500}$	Time	$\bar{r}_{500}$	Time
TFT	pred	**0.03739**	4:33	**0.03726**	8:37	**0.03546**	15:15
TFT	real	0.06971	2:33	0.05815	3:59	0.06391	4:31
DAG	pred	**0.07758**	3:25	**0.07074**	6:04	**0.06499**	10:53
DAG	real	0.48148	1:50	0.77653	2:53	0.69373	3:44
Timer-XL	pred	**0.03805**	1:58	**0.03596**	3:39	**0.03745**	7:58
Timer-XL	real	0.08574	1:17	0.05035	2:05	0.05177	3:57

Generated-context training takes longer in all 27 pairs. At $K = 10$ on MAST, TFT increases from 3:59 to 8:37, and Timer-XL from 2:05 to 3:39. The comparison fixes backbone and stage count, not training time; it therefore does not isolate an equal-compute advantage.

D.3 Returns and costs of additional stages

Table D.3 compares generated-context chains at $K = 10$ and 20. On MAST, TFT and DAG reduce endpoint mean sRMSE by 4.83% and 8.13%, while training time increases by 76.98% and 79.40%. Timer-XL instead worsens by 4.15% with 118.26% more training time. Its $K = 10$ configuration is both faster and more accurate than $K = 20$ under these measurements.

Table D.3: Changes in `pred` from $K = 10$ to $K = 20$. Error reduction uses the $K = 10$ error as denominator; a negative value denotes deterioration. Duration increases use the original minute counts.

Dataset	Backbone	Error reduction	Duration increase
MAST	TFT	4.83092%	76.98259%
MAST	DAG	8.12836%	79.39560%
MAST	Timer-XL	-4.15184%	118.26484%
TEP	TFT	8.06041%	58.29847%
TEP	DAG	21.63079%	77.13568%
TEP	Timer-XL	3.84349%	72.68293%
C-MAPSS	TFT	41.88034%	72.59174%
C-MAPSS	DAG	49.64816%	73.32317%
C-MAPSS	Timer-XL	13.97380%	69.36090%

C-MAPSS shows continued gains: TFT, DAG, and Timer-XL improve by 41.88%, 49.65%, and 13.97% from 10 to 20 stages. TEP improves by 8.06%, 21.63%, and 3.84%, respectively. All generated-context chains improve from 5 to 20 stages, but the intermediate increments vary. These measurements support neither universal diminishing returns nor monotonic realized accuracy. The-

orem A.1 concerns optimal risk in nested realizable classes, not the outcome of greedy finite-data training.

A separate stage-count sweep. Figure D.1 presents a separate MAST/TFT study using endpoint fixed-scale sRMSE at lead 500. Its experimental parameter settings differ from those used in Table 1; the sweep supports within-study scaling comparisons, rather than direct numerical comparisons with the main table. Across the retained counts $K \in \{1, 2, 3, 4, 5, 10, 17\}$, error decreases from 0.1718 to 0.0240, an 86.01% reduction. The improvement from 10 to 17 models is 3.57% (0.0249 to 0.0240). These measurements illustrate smaller gains at the larger retained counts, consistent with the specialization analysis in Appendix A.4; they do not identify an irreducible error floor.

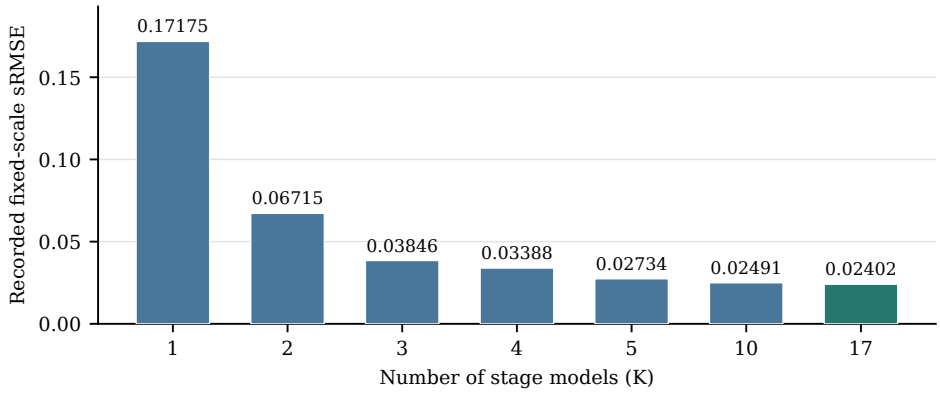


Figure D.1: MAST/TFT stage-count scaling. Experimental parameter settings differ from Table 1; compare stage counts within this sweep. The seven retained configurations show larger early improvements and smaller later gains.

D.4 Public-boundary adaptation

The supplementary MAST trajectory experiments in this subsection use experimental parameter settings that differ from those of the main comparison in Table 2. Consequently, their endpoint errors need not match the table entries. Each adaptive curve is compared with its accompanying fixed-stage baseline within the same experiment; these curves illustrate temporal error behavior and are not pooled with the main results. Table D.4 records boundaries for these trajectory experiments.

The MAST slope-rule experiments construct public boundaries offline on MAST validation data, using the original recent-median and consecutive-confirmation rule. Validation error is read after each additional 10 time points. An initial 50 time points establish a baseline from the preceding four error increments. Two consecutive increments exceeding 1.2 times the recent median trigger a switch. The first flagged prediction is retained and the second probe discarded; the successor trains on the accepted continuous generated history and establishes a new monitoring baseline. These curves isolate the recent-median variant without the slope floor, perturbation margin, or independent candidate-selection rule. The regularized signal analysis in Appendix B and the optional independent selection guarantee are distinguished from this original controller diagnostic.

All three comparisons cover the same 500 future time points, against a fixed 10-model baseline. Table D.4 lists the accepted intervals: TFT uses 6 models, and DAG and Timer-XL each use 7, reducing model counts by 40%, 30%, and 30%. These results establish the feasibility of longer stage reuse across these backbones over this horizon. Whether the resulting accuracy is adequate must be assessed from the errors, not from model count alone.

Error trajectories. The TFT curves are broadly close, but the adaptive curve has pronounced local increases and a slightly higher endpoint (Figure D.2). For DAG, adaptive error is higher over some intermediate intervals, while its displayed endpoint is 0.0813, below the fixed baseline's 0.0828 (Figure D.3). Timer-XL exhibits an early local increase, followed by generally lower adaptive error

Table D.4: Accepted boundaries in the supplementary MAST trajectory experiments, whose parameter settings differ from Table 2. Each boundary is the cumulative number of accepted future time points; successive entries delimit a stage. Model-count reductions are relative to the respective fixed 10-model baseline.

Backbone	K	Fewer models	Accepted boundaries
TFT	6	40%	0, 80, 140, 210, 270, 370, 500
DAG	7	30%	0, 90, 180, 240, 350, 410, 480, 500
Timer-XL	7	30%	0, 100, 160, 250, 360, 420, 490, 500

in the later portion (Figure D.4). These crossings show that fewer models can coexist with local accuracy losses; a single endpoint also cannot replace evaluation over the full horizon.

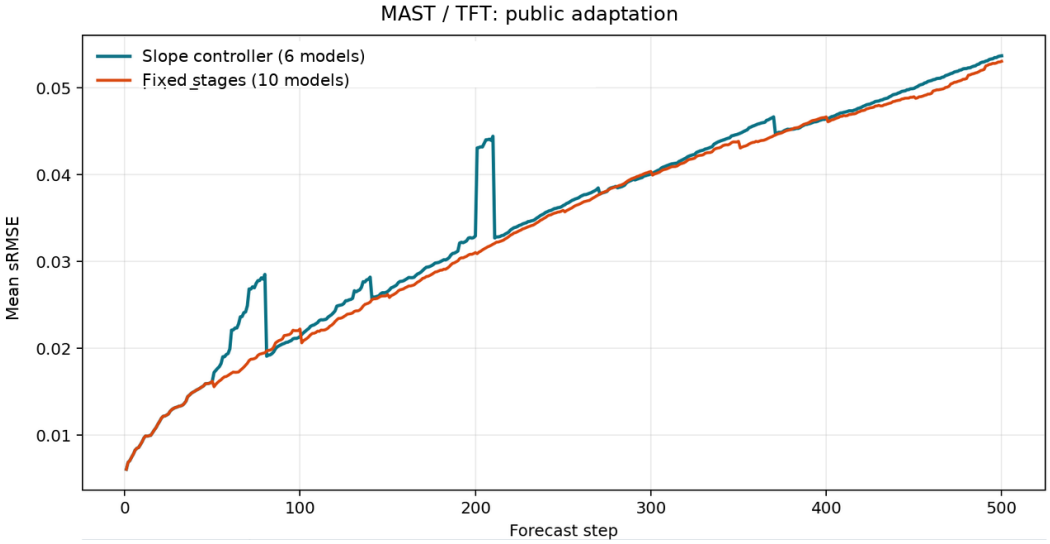


Figure D.2: Adaptive and fixed 10-model TFT curves from the supplementary MAST experiment; parameter settings differ from Table 2. Stage boundaries are given in Table D.4. The adaptive chain uses 6 models but does not have lower error everywhere.

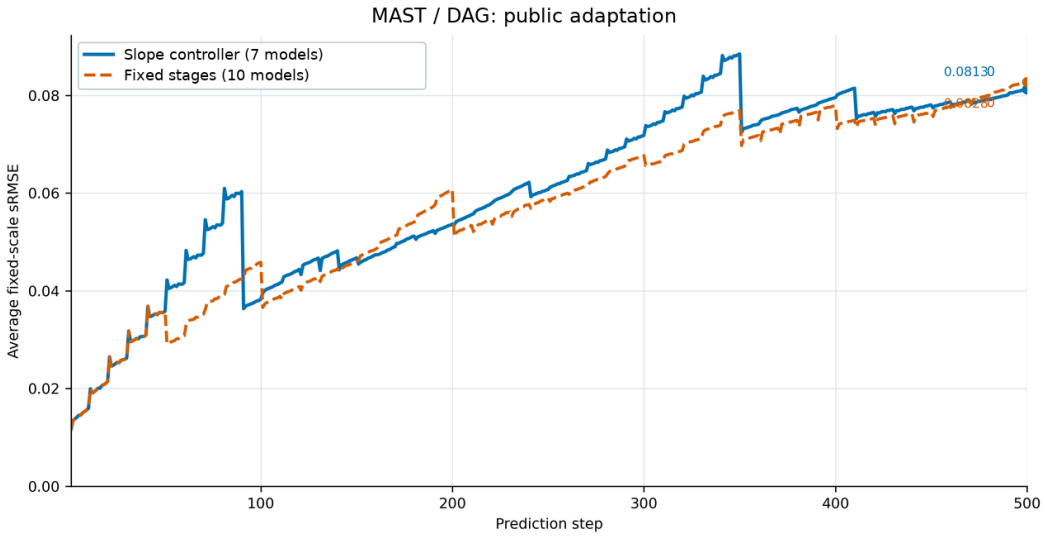


Figure D.3: Adaptive and fixed 10-model DAG curves from the supplementary MAST experiment, using fixed-scale sRMSE; parameter settings differ from Table 2. The adaptive chain uses 7 models; the annotated endpoint advantage does not establish improvement at every intermediate position.

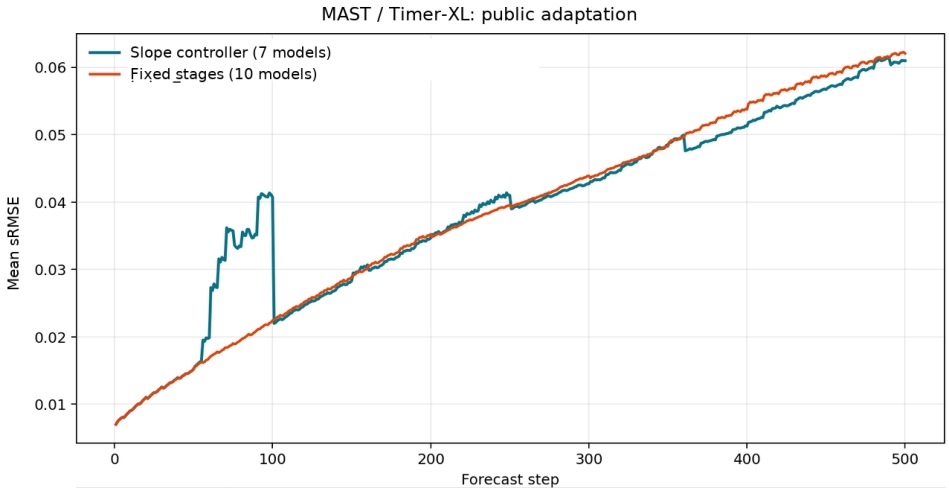


Figure D.4: Supplementary MAST/Timer-XL comparison over the common horizon; parameter settings differ from Table 2. The adaptive chain uses seven models versus ten; its error is locally higher early and generally lower later. Accepted boundaries are given in Table D.4.

D.5 CROSS-DATASET PUBLIC-CONTROLLER TRADEOFFS

Table D.5 summarizes the two public controllers relative to the fixed ten-model baseline, using

$$\Delta m = 100 \left(\frac{m_{\text{adaptive}}}{m_{K=10}} - 1 \right) \%, \qquad m \in \{\bar{r}, q90, q95\}.$$

A negative change is an improvement. All reported configurations are retained: selecting the better controller after examining test errors would not establish a deployable selection rule. Average model use is reported separately from accuracy and is not a substitute for training or inference time.

Table D.5: Public-controller tradeoffs relative to fixed $K = 10$. Changes use the reported Table 2 metrics; negative percentages indicate lower error. $\bar{K}$ is the mean used model count. No timing improvement is inferred.

Dataset	Backbone	Policy	$\bar{K}$	$\Delta\bar{r}$	$\Delta q90$	$\Delta q95$
MAST	TFT	Linear	9	-5.31344%	-7.19798%	-7.38774%
		Slope	6	$+2.82042\%$	-3.43827%	-8.56964%
	DAG	Linear	10	$+0.00000\%$	$+0.00000\%$	$+0.00000\%$
		Slope	7	-3.35162%	-0.43530%	-2.02355%
	Timer-XL	Linear	8	-0.65354%	$+1.52626\%$	$+0.94174\%$
		Slope	7	-1.88831%	-0.17819%	-1.68075%
TEP	TFT	Linear	9	$+8.81350\%$	$+15.21570\%$	$+11.74837\%$
		Slope	3	$+91.62601\%$	$+129.81602\%$	$+124.63973\%$
	DAG	Linear	10	-0.38073%	$+7.97744\%$	$+10.00692\%$
		Slope	3	$+55.00036\%$	$+39.50306\%$	$+48.43275\%$
	Timer-XL	Linear	6	$+13.09449\%$	$+15.76845\%$	$+17.06738\%$
		Slope	8	$+1.85837\%$	$+0.78311\%$	-0.34059%
C-MAPSS	TFT	Linear	8	$+1.23980\%$	$+13.74583\%$	$+22.46820\%$
		Slope	8	-5.43611%	$+0.21642\%$	$+0.25262\%$
	DAG	Linear	5	-11.47682%	-7.25189%	-16.61545%
		Slope	1	$+977.81252\%$	$+879.19771\%$	$+738.29734\%$
	Timer-XL	Linear	4	$+21.11937\%$	$+8.12453\%$	$+17.19448\%$
		Slope	8	-5.85920%	-13.84460%	-3.30897%

MAST. Slope control reduces model use by 30–40%. TFT has a 2.82% higher mean error but lower $q90$ and $q95$, whereas DAG and Timer-XL improve all three error measures. Linear control has lower mean error on TFT and Timer-XL and identical reported errors to fixed ten models on DAG. Improvements in the mean and tail need not coincide: TFT linear control improves the mean and $q90$ relative to fixed ten models, but the fixed five-model configuration still has a lower $q95$.

TEP. Slope control uses three models for TFT and DAG, but mean error rises by 91.63% and 55.00%, accompanied by worse tail errors. Timer-XL slope control uses eight models: mean error rises by 1.86%, $q90$ rises slightly, and $q95$ decreases slightly. Linear control uses $9/10/6$ models for TFT/DAG/Timer-XL. DAG's mean improves by only 0.38% with no model saving, while both tail metrics worsen. TFT and Timer-XL linear control save models but worsen all three error metrics. TEP therefore demonstrates substantial limits to accuracy-preserving adaptation.

C-MAPSS. Slope control uses eight models and improves mean error by 5.44% for TFT and 5.86% for Timer-XL. TFT's two tail metrics increase slightly, while Timer-XL improves both. DAG linear control uses five models and improves mean error by 11.48% and both tail metrics. TFT linear control uses eight models, a 20% reduction from ten, with a 1.24% increase in mean error and 13.75%/22.47% increases in $q90$/$q95$. By contrast, DAG slope control uses one model with 10.78 times the fixed ten-model mean error. These outcomes show that model savings can accompany severe forecast degradation.

The patterns do not support one universally reliable controller. Aggregate error signals can guide useful model reuse, but do not guarantee that a chosen boundary schedule preserves mean or tail accuracy. The population error bounds motivate monitoring; they do not prove the empirical success

of either controller. The tabulated results also do not identify whether a failure originates in threshold calibration, the realized training windows, or model specialization.

D.6 Frozen-chain sample-specific diagnostics on MAST

PredCmp and TrueCmp are evaluated only on MAST, with the same frozen slope-trained chain and five-percent sRMSE comparison rule. They differ in predicted versus future-target errors. Decisions can change subsequent histories, so the comparison concerns complete inference policies.

Table D.6: MAST TrueCmp versus PredCmp. Downward arrows report relative error reductions, $100(1 - m_{\mathrm{TrueCmp}}/m_{\mathrm{PredCmp}})\%$. The final columns give mean model use. TrueCmp requires future targets and is diagnostic only.

Backbone	$\bar{r}\downarrow$	$q90\downarrow$	$q95\downarrow$	$\bar{K}_{\mathrm{Pred}}$	$\bar{K}_{\mathrm{True}}$
TFT	17.50444%	22.93276%	18.27165%	6.65330	5.71350
DAG	24.40602%	21.37432%	23.04471%	6.80980	7.06080
Timer-XL	9.94684%	19.04721%	20.83804%	6.98370	5.57100

TrueCmp improves mean, $q90$, and $q95$ over PredCmp on all three backbones (Table D.6). It uses fewer models on TFT/Timer-XL but more on DAG. Its errors are lowest across all reported TFT/Timer-XL configurations; on DAG, fixed twenty models has lower mean and $q90$, while TrueCmp has lower $q95$. It does not dominate every fixed baseline.

What the comparison establishes. The tested predictors do not recover future-informed routing performance. This does not isolate generalization as the cause: conditional uncertainty, ranking errors, and route-induced shifts may contribute. Accurate risk prediction alone does not guarantee that an absolute-error switching rule selects a better successor. Population bounds neither prescribe individual trajectories nor guarantee that a successor improves on the current model.

Scope of the tail-error evidence. Lower marginal quantiles describe distributional improvements, rather than guaranteed gains for the same difficult cases. The aggregate comparison does not establish predictor calibration or statistical significance. Sample-routing conclusions remain limited to MAST. Even with a prespecified margin, TrueCmp uses future targets; its gains indicate diagnostic potential, not deployable reliability.

E Limitations

Population bounds neither prescribe individual error curves nor prove switching superiority. True-error comparison requires future targets and guarantees no horizon-optimal route. Frozen-chain routing can induce entrance mismatch; gains do not arise from retraining. Predictor generalization and routing costs constrain deployment. Sample-specific evidence is limited to MAST; cross-system routing and margin sensitivity remain unvalidated.