ROAM: TASK-CONDITIONED METRICS FOR FORECASTING OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Can a forecasting task inform how its model should be optimized, without changing the forecasting architecture? We study this question through ROAM (Riemannian Optimization with Adaptive Metrics), which maps six training-derived task statistics to a block-wise positive-definite metric. A slow population controller adapts the metric and a two-signal transition between momentum-based and quasi-Newton updates. The central mechanism is task-conditioned preconditioning on ordinary parameters; explicit orthogonality and positive-definite parameter constraints are separate, optional interventions. Across seven datasets and five backbones, ROAM improves all 35 dataset–backbone MSE aggregates relative to standard training recipes, with approximately 4.8%-7.3% lower macro-average MSE by backbone. Four matched-budget optimizer comparisons and a complete three-component factorial provide more targeted evidence: the metric has the largest average marginal contribution, while switching and outer adaptation also help. A local analysis relates metric alignment and cross-block coupling to conditioning, without assuming that task descriptors guarantee alignment. Together, these results support task information as a useful input to training geometry in the evaluated forecasting regimes.

1 INTRODUCTION

Time-series forecasting is no longer a niche prediction problem; it is a decision-making substrate for modern infrastructures. Applications such as power dispatch, traffic control, financial monitoring, healthcare operations, and climate-responsive planning all rely on forecasts that are not merely accurate on average, but stable across horizons, robust across domains, and efficient under finite training budgets. This practical demand has driven remarkable progress in forecasting model architectures. Transformer-based models, such as Informer (Zhou et al., 2021), Autoformer (Wu et al., 2021), FEDformer (Zhou et al., 2022), Crossformer (Zhang & Yan, 2023), PatchTST (Nie et al., 2023), and iTransformer (Liu et al., 2024b), have successively improved long-range dependency modeling, decomposition, and cross-variate interaction learning. In parallel, strong non-attention alternatives—including DLinear (Zeng et al., 2023), TimesNet (Wu et al., 2023), TimeMixer (Wang et al., 2024), and ModernTCN (Luo & Wang, 2024)—have shown that competitive forecasting does not depend on a single architectural paradigm. More recently, foundation-model directions such as Time-LLM (Jin et al., 2024), Timer (Liu et al., 2024c), TimesFM (Das et al., 2024), MOMENT (Goswami et al., 2024), Chronos (Ansari et al., 2024), and Moirai (Woo et al., 2024) have shifted attention toward transfer, zero-shot forecasting, and broad-domain temporal generalization.

Yet this literature is still dominated by an *architecture-first* view of progress. Most forecasting papers ask how to redesign the temporal encoder, how to improve decomposition, or how to enlarge receptive fields. Far less attention has been paid to a more basic question: **what training geometry should govern forecasting models when tasks differ sharply in noise, seasonality, horizon difficulty, and cross-series structure?** In current practice, highly heterogeneous forecasting tasks are typically optimized in a fixed Euclidean parameter space with a broadly generic optimizer and a manually specified schedule. This convention is computationally convenient, but conceptually restrictive. It implicitly assumes that one geometry, one optimizer regime, and one schedule family are suitable for all forecasting problems.

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

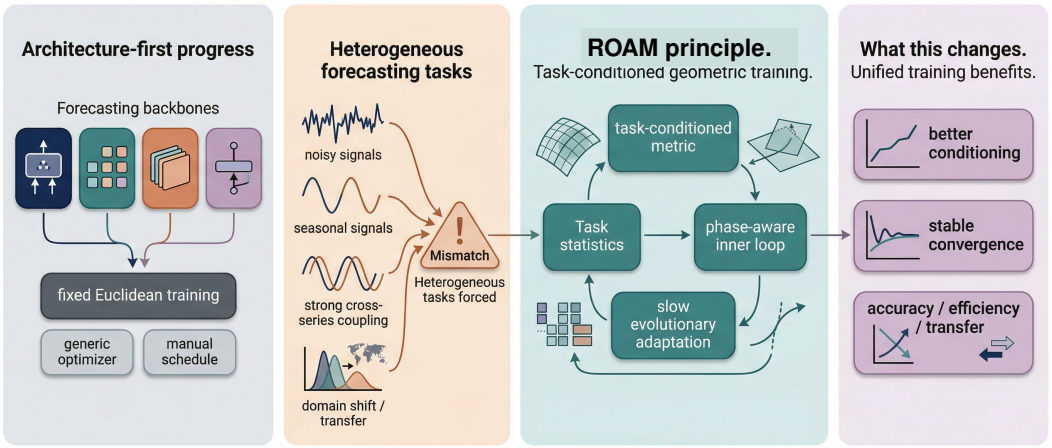

Figure 1: Conceptual motivation for task-conditioned geometric training. Forecasting tasks differ in noise, periodic structure, coupling, and domain; ROAM uses these differences to initialize and adapt training geometry. The benefit icons summarize the mechanisms and evaluation dimensions studied in the paper.

We argue that this assumption is increasingly untenable. Long-horizon forecasting tasks differ substantially in gradient noise, local anisotropy, effective curvature, and sensitivity to hyperparameters (Kingma & Ba, 2015; Foret et al., 2021; Liu et al., 2024a; Bécigneul & Ganea, 2019). A model trained on meteorological variables doesn't induce the same optimization landscape as one trained on other datasets. Even within the same architecture, optimization stiffness can change drastically across datasets and horizons. As forecasting models become stronger, these mismatches are no longer hidden by representational bottlenecks; they emerge directly as slow convergence, seed sensitivity, unstable late-stage training, and brittle transfer. Foundation models sharpen this point further. If forecasting is moving toward universal or cross-domain systems, then the *training principle* should generalize across tasks rather than relying on dataset-specific manual tuning, as emphasized by TimesFM (Das et al., 2024), Chronos (Ansari et al., 2024), MOMENT (Goswami et al., 2024), and Moirai (Woo et al., 2024).

This observation reveals a gap in the current literature. The forecasting community has invested heavily in temporal representation design, but has not developed an equally principled account of *task-aligned optimization*. Meanwhile, the optimization literature already provides strong clues about how such an account might look. Natural gradient methods (Amari, 1998) show that steepest descent depends on geometry rather than coordinates. Matrix-manifold optimization (Absil et al., 2008) provides a rigorous framework for structured parameter spaces through tangent projections, retractions, and manifold-aware first- and second-order methods. Riemannian stochastic optimization (Bonnabel, 2013) and Riemannian adaptive methods (Bécigneul & Ganea, 2019) extend this perspective to noisy large-scale learning. Population-Based Training (Jaderberg et al., 2017) shows that effective schedules can be discovered online rather than fixed in advance. Recent bilevel optimization results on manifolds, including the framework of Han et al. (2024) and the analysis of Li & Ma (2025), indicate that geometry and slow-timescale adaptation can, in principle, be optimized jointly. However, these lines of work remain largely disconnected from mainstream forecasting.

The central claim of this paper is more precise than "geometry matters": *ROAM learns a task-conditioned metric controller that reduces effective optimization anisotropy across heterogeneous forecasting tasks*. Based on this claim, we propose **ROAM**, **R**iemannian **O**ptimization with **A**daptive **M**etrics. ROAM maps task statistics to a slow control variable γ. This variable controls three coupled objects: the block-wise metric Q_γ, the state-triggered transition from a noise-robust first-order phase to a curvature-aware quasi-Newton phase, and the resource-aware outer-loop utility. The resulting system treats geometry, optimizer phase, and budget allocation as one mechanism rather than as separately tuned engineering choices. This framing gives three testable contributions.

- **Mechanism.** We formulate forecasting optimization as task-conditioned metric learning over a product parameter manifold, where descriptors such as SNR, spectral entropy, graph density, and gradient-noise scale generate Q_γ.
- **Theory.** We characterize how local spectral alignment between Q_γ and curvature controls the effective condition number and local contraction factor. We further motivate the phase switch using gradient-noise and curvature-pair reliability.
- **Evidence.** We evaluate matched-budget optimizer tournaments, descriptor ablations, anisotropy diagnostics, and supplementary source–target transfer, in addition to standard forecasting benchmarks. These tests are designed to isolate the contribution of the task descriptor and metric controller, rather than merely whether a larger optimizer stack helps.

Our goal is therefore different from the standard "one more forecasting block" paradigm. We ask what training principle should govern modern forecasting models when tasks vary in noise, spectrum, coupling, horizon length, and optimization stiffness. The answer developed in this paper is that forecasting should be trained with a task-conditioned geometric system whose claims are checked by conditioning, gradient-noise, sharpness, stiffness, efficiency, and transfer diagnostics. Figure 1 shows the modelling logic.

2 Related work

Forecasting advances span temporal and variate tokenization, decomposition, mixing, convolution, and foundation models; these primarily change representations or pretraining. ROAM instead conditions the training rule on task statistics. Adaptive optimizers and structured preconditioners already modify update geometry, including AdamW, K-FAC, and Shampoo (Loshchilov & Hutter, 2019; Martens & Grosse, 2015; Gupta et al., 2018). Our distinction is explicit forecasting-task conditioning, not the invention of preconditioning. Manifold methods provide constraint-aware updates, learned optimizers generate update rules from experience, and PBT adapts hyperparameters online (Absil et al., 2008; Bécigneul & Ganea, 2019; Andrychowicz et al., 2016; Jaderberg et al., 2017). ROAM retains a prescribed metric family and uses derivative-free population control; it is not a manifold hypergradient algorithm. Appendix K gives the full discussion of forecasting backbones, objectives, geometric optimization, outer adaptation, and manifold bilevel methods, including their relationship to our formulation.

3 Task-conditioned metric control

Figure 2 summarizes the two-timescale controller: task statistics initialize the geometry, the inner loop updates the forecasting backbone, and validation feedback adapts the slow controls. The metric acts on the training update; it does not add a forecasting layer.

3.1 Task, blocks, and the metric interface

Let $f_\theta : \mathbb{R}^{T\times d} \to \mathbb{R}^{H\times d}$ forecast H steps from a context of length T. Training minimizes the average forecasting loss $\mathcal{L}_{\mathrm{tr}}(\theta)$, with regularization fixed within each comparison. Validation selects controls and checkpoints; test targets enter neither descriptor construction nor control selection. Partition $\theta = (\theta^{(1)}, \dots, \theta^{(B)})$ by backbone modules before comparing optimizers. For attention models these groups follow input/patch projections, attention projections, feed-forward sublayers, and the head. Mixer and convolution models follow their stages; DLinear follows its trend and seasonal maps and any separate output map. Bias and normalization parameters attach to the corresponding parent group. Block diagonality approximates curvature storage and solves; it does not assert statistical or functional independence.

For ordinary vectorized blocks of dimensions n_b, define the metric and its gradient by

$$G_{\gamma,\theta} = \mathrm{blkdiag}(\alpha_1 Q_1, \dots, \alpha_B Q_B), \qquad g_{\gamma,\theta}(u,v) = u^\top G_{\gamma,\theta} v, \tag{1}$$

$$Q_b = (\lambda_b + \delta) I + \omega_b \widehat{F}_b + (1-\omega_b) D_b, \qquad \mathrm{grad}_g\, \mathcal{L}_{\mathrm{tr}} = G_{\gamma,\theta}^{-1} \nabla \mathcal{L}_{\mathrm{tr}}. \tag{2}$$

Here $\alpha_b, \lambda_b > 0$, $\omega_b \in [0,1]$, and $\delta > 0$. The positive-semidefinite proxy $\widehat{F}_b$ represents Gauss–Newton/Fisher-type curvature; conceptually it is $J_b^\top J_b / |\mathcal{B}|$ for a specified block-output Jacobian.

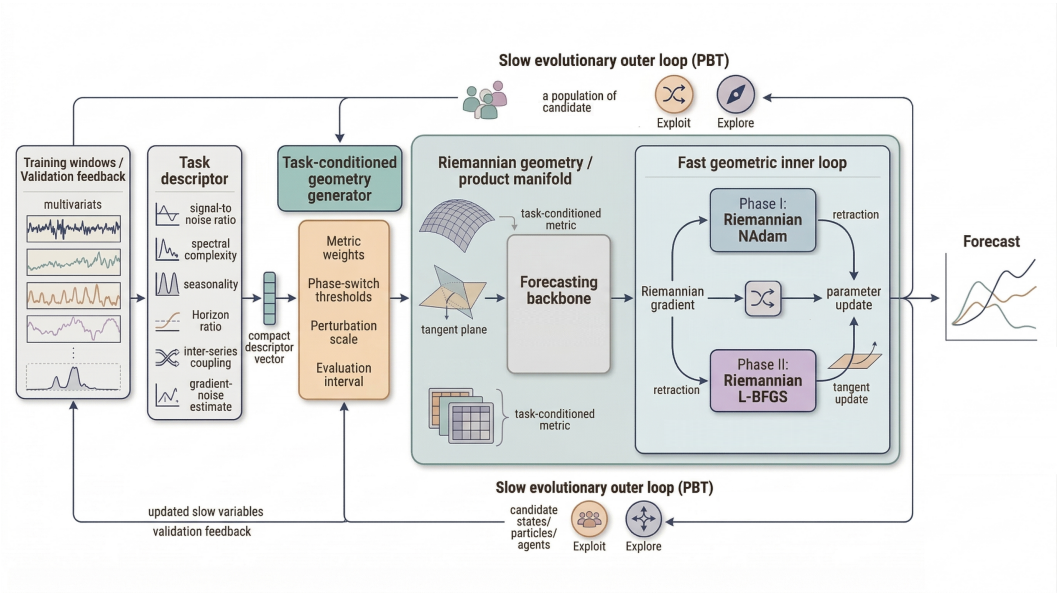

Figure 2: ROAM framework. Training-derived descriptors initialize metric and switching controls. Fast momentum/quasi-Newton updates train the backbone, while a slow population loop uses validation feedback to adapt controls. The product-manifold view includes ordinary parameter blocks; Stiefel and SPD parameter constraints are configuration-dependent, not mandatory for every backbone.

The second proxy is $D_b = \mathrm{diag}(\mathrm{EMA}[(\nabla_b \ell)^2])$. It is a nonnegative gradient-second-moment proxy, not the diagonal of the exact Hessian. Consequently $Q_b \succeq (\lambda_b + \delta)I$. The metric changes the differential-to-direction map while leaving f_θ unchanged.

The matrix expression specifies an operator, not a requirement to materialize dense $n_b \times n_b$ arrays. Its implementation requires a storage approximation, update cadence, and solve rule; these determine the accuracy–cost trade-off. Appendix D discusses the numerical configuration and computational trade-offs.

3.2 From task statistics to slow controls

The descriptor is

$$c = [\mathrm{snr}, \mathrm{sent}, \mathrm{seas}, H/T, \mathrm{dens}, \mathrm{gns}]^\top, \qquad z^{(0)} = W\widetilde{c} + b, \quad \gamma^{(0)} = \mathcal{P}(z^{(0)}), \tag{3}$$

where $\widetilde{c}$ denotes standardized descriptors and $\mathcal{P}$ enforces the control domains. Positive entries use softplus, mixture weights use sigmoid, and evaluation intervals are positive integers. The first five descriptors summarize signal strength, spectral concentration, seasonality, horizon relative to context, and cross-series coupling from training data and the task definition. Only gns uses model initialization, through two independent mini-batch gradients at the same θ_0. Appendix B gives the estimators and data dependencies; Appendix G reports estimator sensitivity. These six entries are an auditable control input, not a claimed sufficient statistic of all forecasting tasks.

The metric family $(\alpha_b, \lambda_b, \omega_b)_{b=1}^B$ governs scaling and curvature mixing. The thresholds (τ_χ, τ_ρ) govern phase transitions. Exploration scale σ and interval T_{eval} govern the slow loop; time and memory weights govern resource preferences. Learning-rate schedule shape is not another evolved coordinate. The affine initializer (W, b) and descriptor standardization are distinct from the per-task population update (Appendix D).

A scalar learning rate changes update magnitude uniformly, whereas the task metric selects relative block scales and curvature mixtures. Within-block proxies supply directional information. Appendix A.2 gives an exact quadratic illustration of this distinction.

3.3 INNER UPDATES AND STATE-TRIGGERED SWITCHING

Write $g_t = \operatorname{grad}_g \ell_{\mathcal{B}_t}(\theta_t)$ for the mini-batch metric gradient and $\langle \cdot, \cdot \rangle_t$ for the current metric inner product. Phase I combines transported momentum and a scalar second moment per block in a NAdam-style update (Dozat, 2016; Bécigneul & Ganea, 2019); Appendix C.1 specifies the recurrence and retraction. Phase II uses $p_t = -\mathcal{H}_t g_t$ from the latest M accepted L-BFGS pairs and an Armijo backtracking step. Both members of each pair must be represented in the same tangent space and evaluated under the same metric. With s, y denoting such a displacement and transported gradient difference, retain only pairs satisfying

$$\langle s, y \rangle_t \geq c_{\mathrm{curv}} \|s\|_t^2 > 0. \tag{4}$$

Transport, metric changes, and the precise scope of positive-definiteness preservation are detailed in Appendix C.

At a diagnostic check, draw two independent mini-batch gradients $g_t^{[1]}, g_t^{[2]}$ under the same metric and compute

$$\chi_t = \frac{\left\| g_t^{[1]} - g_t^{[2]} \right\|_t^2}{\left\| (g_t^{[1]} + g_t^{[2]})/2 \right\|_t^2 + \varepsilon}, \qquad \rho_t = \frac{\langle s, y \rangle_t}{\|s\|_t^2 + \varepsilon}. \tag{5}$$

Enter Phase II when $\chi_t \leq \tau_\chi$ and $\rho_t \geq \tau_\rho$ hold for $q = 3$ consecutive checks; return to Phase I after q consecutive failures. Without a valid pair, entry is not certified. The rule provides hysteresis and a directional-curvature safeguard, not a statistical certificate that the full Hessian is positive definite.

3.4 OUTER ADAPTATION AND CONSTRAINT SCOPE

A population of $N_{\mathrm{pop}} = 4$ shares the backbone and initial weights but perturbs the slow controls. After an evaluation interval, individuals are scored using

$$J_i^{(r)} = \mathcal{L}_{\mathrm{val}}(\theta_i^{(r)}) + \lambda_{\mathrm{time}} \frac{C_i^{\mathrm{time}}}{C_0^{\mathrm{time}}} + \lambda_{\mathrm{mem}} \frac{C_i^{\mathrm{mem}}}{C_0^{\mathrm{mem}}}. \tag{6}$$

The top quarter supplies replacements for the bottom quarter, preserving the best individual. Multiplicative perturbations explore continuous controls, with domain projection and valid integer rounding. A meaningful population ranking requires common resource weights and normalizers within that ranking; candidate-specific discounts would compare different objectives. Algorithm 1 in Appendix C specifies the complete loop; Appendix D gives the numerical configuration.

An *SPD metric operator* in Eq. (1) is always present when the task metric is enabled. An *SPD parameter block*, by contrast, is a trainable matrix constrained to $\mathbb{S}_{++}$. It is activated only in a configuration with an eligible positive-definite trainable parameter. The same principle applies to Stiefel constraints on orthogonal matrices. Adding such a constraint to an otherwise unconstrained weight is an additional intervention, even if tensor shapes do not change. We therefore report ordinary-weight metric effects, Stiefel effects, and the separate SPD-subset result with distinct scopes (Section 5.3); we do not infer backbone eligibility from a preconditioner's positive definiteness.

4 WHAT ALIGNMENT CAN EXPLAIN

The analysis isolates the metric mechanism from the full adaptive optimizer. Let $K = \nabla^2 \mathcal{L}_{\mathrm{tr}}(\theta_\star) \succ 0$ at a nondegenerate local minimum in local coordinates. Freeze the slow controls and let $G_\star \succ 0$ be the metric there. Define $A = G_\star^{-1/2} K G_\star^{-1/2}$ and $\kappa(A) = \lambda_{\max}(A)/\lambda_{\min}(A)$.

Table 1: Forecasting breadth. Macro means over seven datasets and four horizons; standard recipe → ROAM. Relative MSE gains are recomputed from the dataset-level values before rounding the macro means. Per-dataset seed deviations are in Appendix E.

Backbone	MSE	MAE	MSE gain
iTransformer	$0.334 \to 0.313$	$0.366 \to 0.349$	6.2%
PatchTST	$0.305 \to 0.290$	$0.349 \to 0.336$	4.8%
TimeMixer	$0.332 \to 0.313$	$0.363 \to 0.348$	5.6%
ModernTCN	$0.315 \to 0.299$	$0.355 \to 0.342$	5.0%
DLinear	$0.334 \to 0.310$	$0.374 \to 0.355$	7.3%

Proposition 1 (Local metric alignment). *If $aK \preceq G_\star \preceq bK$ for $0 < a \le b$, then*

$$\frac{1}{b} I \preceq A \preceq \frac{1}{a} I, \qquad \kappa(A) \le b/a. \tag{7}$$

For metric-gradient descent on the local quadratic, the optimal constant step $\eta_\star = 2/(\lambda_{\min}(A) + \lambda_{\max}(A))$ has contraction factor $r_\star = (\kappa(A) - 1)/(\kappa(A) + 1)$ in the $G_\star$ norm. For a loss with locally Lipschitz Hessian, a locally Lipschitz uniformly SPD metric, and a smooth retraction, the corresponding local update satisfies

$$\|e_{t+1}\|_{G_\star} \le r_\star \|e_t\|_{G_\star} + O(\|e_t\|_{G_\star}^2).$$

Congruence of the spectral sandwich proves Eq. (7); whitening the error yields the contraction (Appendix A). The condition $b/a < \kappa(K)$ is sufficient, not necessary, for a better optimal local factor than identity-metric descent. The special case $(1 - \epsilon)K \preceq G_\star \preceq (1 + \epsilon)K$ gives $\kappa(A) \le (1 + \epsilon)/(1 - \epsilon)$. Neither assumption is guaranteed by an affine descriptor generator.

Block diagonality imposes a further limitation. Set $D = \mathrm{blkdiag}(K_{11}, \ldots, K_{BB})$ and $\nu = \left\| D^{-1/2}(K - D)D^{-1/2} \right\|_2$. If $\nu < 1$ and $aD \preceq G_\star \preceq bD$, then

$$\kappa(A) \le \frac{b}{a} \frac{1 + \nu}{1 - \nu}. \tag{8}$$

This derived bound (Appendix A.3) separates within-block alignment from neglected cross-block coupling. It explains why module-aligned blocks can help without asserting independence, and why matching diagonal blocks alone may be insufficient.

Stochasticity also remains after preconditioning. For a frozen quadratic and unbiased gradient noise ξ_t, the conditional squared-error recursion includes $\eta^2 \operatorname{tr}(G_\star^{-1} \operatorname{Cov}(\xi_t \mid e_t))$. A useful metric must therefore balance alignment, noise amplification, and cost. The switch probes this balance empirically; the analysis does not prove convergence of NAdam, stochastic L-BFGS, metric-estimation dynamics, or the population process.

5 Experiments

5.1 Protocol and forecasting breadth

We evaluate ETTh1, ETTh2, ETTm1, ETTm2, Weather, ECL, and Exchange, with horizons $\{96, 192, 336, 720\}$ and five backbones. The reported protocol uses chronological splits, training-only z-score normalization, normalized-target MSE/MAE, and 10 seeds. Architecture, width/depth, look-back, normalization, batch size, stopping rules, and seed pool are fixed within comparisons. The broad benchmark compares each backbone's standard recipe with ROAM. The stronger optimizer study covers four selected task–backbone–horizon settings; these are different claims.

ROAM lowers MSE in all 35 dataset–backbone aggregates (Table 1 and Appendix E). The mean of per-backbone relative gains is 6.8%, 6.3%, 6.1%, and 6.4% on the four ETT datasets, 5.1% on Weather, and 4.4% on ECL. Exchange varies from 2.1% to 7.6% by backbone. These results establish breadth against the standard recipes; they do not establish a tuned-optimizer win at every individual horizon. We do not infer a cross-dataset anisotropy–gain correlation from dataset names or aggregate errors.

Table 2: Optimizer comparison, test MSE. A: ETTh1-96/iTransformer; B: ETTm1-336/PatchTST; C: Weather-192/TimeMixer; D: Exchange-336/DLinear. Results are means over 10 matched seeds; ROAM deviations appear in Table 4.

Optimizer/control	A	B	C	D
Lion	.379	.357	.243	.378
SAM	.371	.354	.241	.375
Sophia-G	.366	.349	.238	.371
PBT-only	.370	.352	.240	.373
Riemannian-only	.365	.347	.237	.369
NAdam–L-BFGS, no task metric	.363	.345	.236	.367
ROAM	**.358**	**.341**	**.232**	**.360**

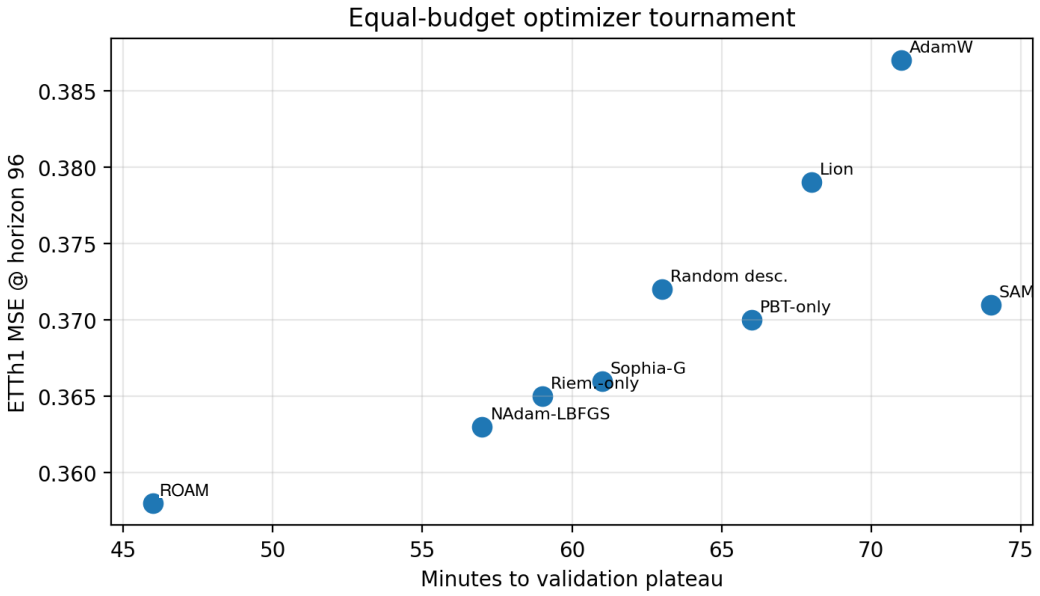

Figure 3: Optimizer comparison on ETTh1-96. Each point shows test MSE and minutes to validation plateau under the matched-budget protocol. Plateau time is a convergence diagnostic, not total population compute. Table 2 extends the accuracy comparison to four settings.

5.2 Four matched-budget optimizer comparisons

Table 2 compares stronger alternatives across four regimes. The protocol reports matched wall-clock/GPU-hour caps, validation-query opportunity, and seeds. On ETTh1-96, each method has 28 validation calls; measured GPU-hours range from 1.13 to 1.22. Matching the budget protocol does not imply identical consumed resources; Appendix H distinguishes per-run measurements from population-wide costs.

ROAM has the lowest reported mean in all four settings. Relative to the no-task-metric hybrid, the absolute reductions are 0.005, 0.004, 0.004, and 0.007 MSE, corresponding to 1.4%, 1.2%, 1.7%, and 1.9%. This comparator is especially informative because hybrid optimization alone does not recover the full result. Figure 3 places the ETTh1-96 comparison on an accuracy–plateau-time plane: ROAM combines the lowest displayed error with the earliest plateau. These are mean-performance comparisons, not paired significance tests.

5.3 Attribution: metric, switching, and outer adaptation

For each component, the marginal gain averages its four on/off contrasts with the other two settings fixed (Table 3). The metric has the largest mean effect, but metric-only training (.323) does not match

Table 3: Complete 2^3 factorial. M: task metric; S: state-triggered switch; O: outer adaptation. Means over ETTh1-192, ETTm1-336, and Weather-192. Effects are calculated from the reported rounded cell means, without inferred uncertainty.

M	S	O	Mean MSE
0	0	0	.333
1	0	0	.323
0	1	0	.329
0	0	1	.327
1	1	0	.318
1	0	1	.319
0	1	1	.324
1	1	1	**.312**

Component	Mean marginal gain
Metric	.0103
Switching	.0048
Outer adaptation	.0053

Table 4: Reported multi-seed dynamics in the four settings of Table 2. MSE and first stable switch are mean $\pm$ seed standard deviation over 10 seeds. Conditioning reductions are normalized proxy summaries, not certified bounds on the full Hessian.

Setting	ROAM MSE	Conditioning reduction	Plateau (min) AdamW $\rightarrow$ ROAM	Switch step
A	$.358 \pm .006$	72%	$71 \rightarrow 46$	860 ± 74
B	$.341 \pm .005$	63%	$58 \rightarrow 39$	940 ± 81
C	$.232 \pm .004$	49%	$43 \rightarrow 31$	780 ± 69
D	$.360 \pm .010$	57%	$52 \rightarrow 36$	910 ± 95

the full system (.312). Adding the metric gives .010 improvement without either other component and .012 when both are active. Switching and outer adaptation also improve every matched cell. This pattern supports complementary roles rather than an additive decomposition. Appendix F provides setting-level ablations and distinguishes the separately reported fixed-identity-plus-PBT control.

Metric versus constrained parameters. With switching and outer adaptation fixed, the geometry decomposition gives identity .324, task metric on Euclidean blocks .313, and task metric plus eligible Stiefel blocks .312 (Table 7). Thus .011 of the .012 absolute decrease in this comparison occurs without the Stiefel intervention. Separately, the SPD-eligible subset comprises the evaluated ModernTCN and DLinear configurations and reports .324 without SPD activation and .310 with activation. That is a 4.3% within-subset reduction, *not* a further .002 improvement over the three-setting .312 row. This is a pooled effect for the two evaluated configurations, not an individual effect estimate for either backbone. Appendix D.4 details the distinction between metric and parameter-constraint activation.

5.4 Descriptor evidence and optimization dynamics

Randomizing the descriptor raises the three-setting mean from .312 to .328. A learned encoder gives .313; adding four statistics gives .311; a hand-crafted-plus-learned residual gives .310. The validation-assisted oracle diagnostic also gives .310, but it is not part of the deployable training-only descriptor. The .001–.002 gaps among the richer descriptors do not establish equivalence or superiority without paired uncertainty. Appendix F reports all six leave-one-out and replacement-descriptor results (Table 15), together with setting-level MSE/MAE.

The largest removal effects concern initial gradient disagreement (.319) and graph density (.317). Estimator perturbations retain .312–.314 MSE; initialization and bootstrap checks are in Appendix G.

The trajectories in Figure 4 show how the validation error and estimated conditioning evolve, complementing the multi-seed summaries in Table 4. Appendix J, Figure 9, separates normalized conditioning, noise, sharpness, and stiffness diagnostics. All four multi-seed summaries show earlier validation plateaus and lower normalized conditioning proxies. Switch times vary across regimes

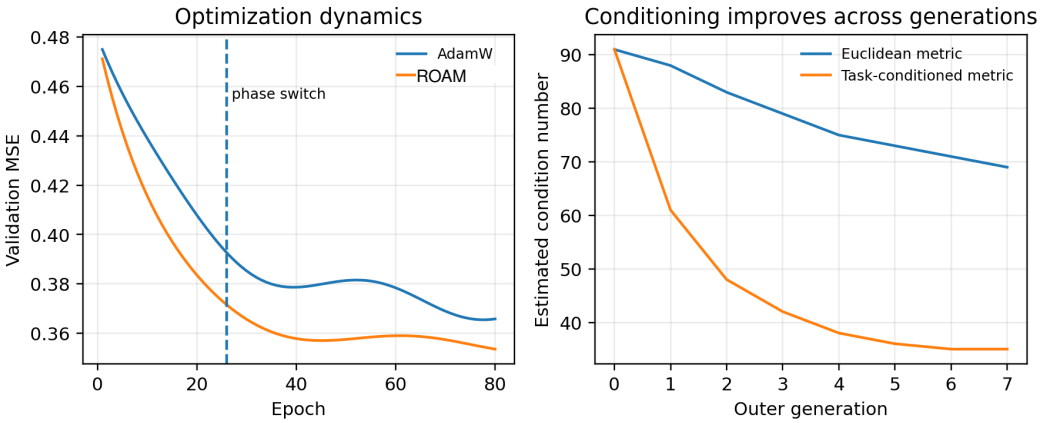

Figure 4: Optimization trajectories. Left: validation MSE versus epoch, with the phase transition marked by the dashed line. Right: estimated condition number versus outer generation. These trajectories illustrate the dynamics; the four-setting, 10-seed summaries are reported separately in Table 4.

and seeds, as expected for a state-based rule. These summaries are consistent with the proposed mechanism, although a conditioning proxy is not a direct test of the spectral sandwich in Proposition 1.

5.5 Accuracy–cost trade-off and scope

On ETTh1-96/iTransformer, ROAM increases relative epoch time from 1.00 to 1.19 and peak GPU memory from 16.1 to 17.4 GB. Its reported plateau time decreases from 71 to 46 minutes (35.2%), with MSE .387 to .358 and reported GPU-hours 1.18 for both methods. Epoch overhead, plateau time, and total budget are different quantities; the last figure is not derived by multiplying the first two. Appendix H reports the full cost table and the corresponding frontier in Figure 5.

Sensitivity tests support a trade-off rather than a unique optimum. Increasing population size from 4 to 6 changes mean MSE .312 to .311 while increasing relative epoch time 1.19 to 1.31. At half, default, and twice the base learning rate, the reported relative MSE gains are 5.1%, 5.9%, and 5.4%. L-BFGS memory, hysteresis, and threshold sensitivity are tabulated in Appendix G. Supplementary source–target results and their cell-wise diagnostic are given in Appendix I (Figure 6); representative forecast traces and descriptor ablations appear in Figures 7 and 8. Our main claims concern training within the evaluated forecasting tasks rather than zero-shot forecasting.

6 Discussion and conclusion

The strongest supported conclusion is that explicit task information helps the tested training system choose useful ordinary-parameter metrics. The broad benchmark demonstrates improvements over standard recipes, while the factorial and no-metric hybrid controls identify a substantial contribution beyond auxiliary optimization machinery. Explicit Stiefel/SPD parameter constraints are separate, configuration-dependent interventions; they are not the explanation for all gains.

The present evidence remains limited to long-horizon multivariate forecasting and four strong-optimizer settings. The descriptor and curvature proxies may miss distribution shift or cross-block coupling; the local theory assumes alignment and a nondegenerate minimum and does not characterize the full adaptive process. The aggregate comparisons characterize mean accuracy, not paired statistical significance or identical population-wide resource use. Further comparisons with structured preconditioners would test whether task conditioning remains beneficial under richer curvature approximations. Within the evaluated scope, ROAM supports treating task-conditioned geometry as a training design choice alongside architecture and preprocessing.

AI USE STATEMENT

In this work, we used generative AI tools during manuscript preparation for editing and polishing. We have not used generative AI tools for dataset processing in this process, and translation and qualitative analysis are not applicable. Additionally, we used generative AI tools for literature/reference checks, and figure-label editing. We have reviewed all AI-assisted work. Checks covered reported numbers, citations, mathematical consistency, compilation, and page rendering. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies optimization using existing forecasting benchmarks and does not introduce new human-subject data collection. Improved benchmark accuracy does not guarantee calibrated uncertainty, fairness, or safe deployment. Forecasting used in infrastructure or other consequential settings requires application-specific validation and human oversight. Additional optimizer search also consumes resources; the cost analysis distinguishes per-epoch overhead from total compute.

REPRODUCIBILITY STATEMENT

Appendix A gives complete proofs of the stated mathematical results. Appendices B and C define the descriptor and geometric operations. Appendix D describes the numerical configuration and implementation dependencies. Appendices E–I provide detailed numerical results and aggregation conventions.

REFERENCES

P.-A. Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2008.

Shun-ichi Amari. Natural Gradient Works Efficiently in Learning. *Neural Computation*, 10(2): 251–276, 1998.

Marcin Andrychowicz, Misha Denil, Sergio Gómez, Matthew W. Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando de Freitas. Learning to learn by gradient descent by gradient descent. In *Advances in Neural Information Processing Systems*, volume 29, 2016. URL https://papers.nips.cc/paper_files/paper/2016/hash/fb87582825f9d28a8d42c5e5e5e8b23d-Abstract.html.

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.

Vincent Arsigny, Pierre Fillard, Xavier Pennec, and Nicholas Ayache. Log-Euclidean metrics for fast and simple calculus on diffusion tensors. *Magnetic Resonance in Medicine*, 56(2):411–421, 2006. doi: 10.1002/mrm.20965.

Gary Bécigneul and Octavian-Eugen Ganea. Riemannian Adaptive Optimization Methods. In *International Conference on Learning Representations*, 2019.

Silvère Bonnabel. Stochastic Gradient Descent on Riemannian Manifolds. *IEEE Transactions on Automatic Control*, 58:2217–2229, 2013.

Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, Yifeng Lu, and Quoc V. Le. Symbolic Discovery of Optimization Algorithms. In *Advances in Neural Information Processing Systems*, volume 36, pp. 4925–4964, 2023.

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, 2024.

Timothy Dozat. Incorporating Nesterov Momentum into Adam. In *International Conference on Learning Representations, Workshop Track*, 2016. URL https://openreview.net/pdf?id=OM0jvwB8jIp57ZJjtNEZ.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-Aware Minimization for Efficiently Improving Generalization. In *International Conference on Learning Representations*, 2021.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. MOMENT: A Family of Open Time-series Foundation Models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 16115–16152, 2024.

Vineet Gupta, Tomer Koren, and Yoram Singer. Shampoo: Preconditioned Stochastic Tensor Optimization. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1842–1850, 2018. URL https://proceedings.mlr.press/v80/gupta18a.html.

Andi Han, Bamdev Mishra, Pratik Jawanpuria, and Akiko Takeda. A Framework for Bilevel Optimization on Riemannian Manifolds. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Max Jaderberg, Valentin Dalibard, Simon Osindero, Wojciech M. Czarnecki, Jeff Donahue, Ali Razavi, Oriol Vinyals, Tim Green, Iain Dunning, Karen Simonyan, Chrisantha Fernando, and Koray Kavukcuoglu. Population Based Training of Neural Networks. *arXiv preprint arXiv:1711.09846*, 2017.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. Time-LLM: Time Series Forecasting by Reprogramming Large Language Models. In *International Conference on Learning Representations*, 2024.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*, 2015.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling Long- and Short-Term Temporal Patterns with Deep Neural Networks. In *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 95–104, 2018.

Jiaxiang Li and Shiqian Ma. Riemannian Bilevel Optimization. *Journal of Machine Learning Research*, 26(18):1–44, 2025. URL https://www.jmlr.org/papers/v26/24-0397.html.

Bryan Lim, Sercan Ö. Arik, Nicolas Loeff, and Tomas Pfister. Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting. *International Journal of Forecasting*, 37(4): 1748–1764, 2021.

Hong Liu, Zhiyuan Li, David Hall, Percy Liang, and Tengyu Ma. Sophia: A Scalable Stochastic Second-order Optimizer for Language Model Pre-training. In *International Conference on Learning Representations*, 2024a.

Yong Liu, Chenyu Li, Jianmin Wang, and Mingsheng Long. Koopa: Learning Non-stationary Time Series Dynamics with Koopman Predictors. In *Advances in Neural Information Processing Systems*, volume 36, pp. 59287–59311, 2023.

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 *International Conference on Learning Representations*, 2024b.

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, 2024c.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*, 2019.

Donghao Luo and Xue Wang. ModernTCN: A Modern Pure Convolution Structure for General Time Series Analysis. In *International Conference on Learning Representations*, 2024.

James Martens and Roger Grosse. Optimizing Neural Networks with Kronecker-factored Approximate Curvature. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2408–2417, 2015. URL https://proceedings.mlr.press/v37/martens15.html.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *International Conference on Learning Representations*, 2023.

Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 2 edition, 2006. doi: 10.1007/978-0-387-40065-5.

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.

Huiqiang Wang, Jian Peng, Feihu Huang, Jince Wang, Junhui Chen, and Yifei Xiao. MICN: Multi-scale Local and Global Context Modeling for Long-term Series Forecasting. In *International Conference on Learning Representations*, 2023.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y. Zhang, and Jun Zhou. TimeMixer: Decomposable Multiscale Mixing for Time Series Forecasting. In *International Conference on Learning Representations*, 2024.

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, 2024.

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.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. In *International Conference on Learning Representations*, 2023.

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.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer Utilizing Cross-Dimension Dependency for Multivariate Time Series Forecasting. In *International Conference on Learning Representations*, 2023.

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.

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, 2022.

A LOCAL ANALYSIS AND COMPLETE PROOFS

The results below concern explicitly stated local models. They explain how metric alignment can affect gradient descent and what a positive-curvature safeguard guarantees. They are not claims that a descriptor necessarily produces alignment or that every step of the stochastic, momentum-based, population-adapted algorithm contracts. Classical preconditioned optimization and matrix-manifold references provide the underlying framework (Nocedal & Wright, 2006; Absil et al., 2008).

A.1 PROOF OF PROPOSITION 1

From $aK \preceq G_\star \preceq bK$, congruence by $G_\star^{-1/2}$ gives

$$aG_\star^{-1/2} K G_\star^{-1/2} \preceq I \preceq bG_\star^{-1/2} K G_\star^{-1/2}.$$

Hence $b^{-1}I \preceq A \preceq a^{-1}I$ and $\kappa(A) \le b/a$. For the quadratic $q(e) = \frac{1}{2}e^\top Ke$, metric-gradient descent is

$$e_{t+1} = (I - \eta G_\star^{-1} K)e_t.$$

Let $z_t = G_\star^{1/2} e_t$. Then $z_{t+1} = (I - \eta A)z_t$ and

$$\|e_{t+1}\|_{G_\star} \le \max_j |1 - \eta\lambda_j(A)| \, \|e_t\|_{G_\star}.$$

The endpoint magnitudes are balanced at $\eta_\star = 2/(\lambda_{\min}(A) + \lambda_{\max}(A))$. At this step the factor is $(\lambda_{\max}(A) - \lambda_{\min}(A))/(\lambda_{\max}(A) + \lambda_{\min}(A)) = r_\star$. This factor is the worst-case one-step factor in the metric norm, not an equality for every starting vector.

For the nonlinear statement, choose a local chart with $\theta_\star$ at the origin. Local Lipschitz continuity of the Hessian implies

$$\nabla\mathcal{L}_{\mathrm{tr}}(e) = Ke + O(\|e\|^2).$$

Uniform positive definiteness and local Lipschitz continuity of the metric imply $G(e)^{-1} = G_\star^{-1} + O(\|e\|)$. Therefore

$$G(e)^{-1}\nabla\mathcal{L}_{\mathrm{tr}}(e) = G_\star^{-1}Ke + O(\|e\|^2).$$

A smooth retraction satisfies, in these coordinates, $R_e(v) = e + v + O(\|v\|^2)$ uniformly in a sufficiently small neighborhood. Substituting $v = -\eta_\star G(e)^{-1}\nabla\mathcal{L}_{\mathrm{tr}}(e)$ yields

$$e_{t+1} = (I - \eta_\star G_\star^{-1} K)e_t + O(\|e_t\|^2).$$

Finite-dimensional norm equivalence and the preceding quadratic bound give the asserted local inequality. Only the first-order retraction property and bounded local second derivatives are needed here; a QR retraction need not be declared second-order.

For a merely C^2 loss without locally Lipschitz Hessian, the corresponding Taylor remainder is generally $o(\|e\|)$, not necessarily $O(\|e\|^2)$. For an overparameterized model with a singular Hessian, the nondegenerate-minimum proposition does not apply directly. A result restricted to a positive-curvature subspace would require specifying that subspace and its invariance under the update. Finally, the comparison of $r_\star$ with identity-metric descent compares their optimal local linear factors, not the realized convergence of arbitrary learning-rate schedules.

A.2 SCALAR VERSUS DIRECTIONAL PRECONDITIONING

What a metric changes. A scalar learning rate rescales every direction equally, whereas G^{-1} can alter their relative motion. For a two-dimensional quadratic with $K = \mathrm{diag}(k_1, k_2)$, $k_1 \ge k_2 > 0$, and $G = \mathrm{diag}(q_1, q_2)$, the effective eigenvalues are k_1/q_1 and k_2/q_2. Their ratio is

$$\kappa(G^{-1/2}KG^{-1/2}) = \frac{\max\{k_1/q_1, k_2/q_2\}}{\min\{k_1/q_1, k_2/q_2\}}.$$

A scalar metric $G = qI$ leaves the condition number k_1/k_2 unchanged, while $G = K$ makes it one. Thus global step-size tuning and directional metric selection address different aspects of the same update. This example is an exact quadratic calculation, not an assumption that a forecasting network has diagonal curvature. In ROAM, the task controls select relative block scales and curvature mixtures; the within-block proxies supply directional information. The empirical question is whether the descriptors choose those controls more usefully than a task-agnostic or randomized alternative, which motivates the controls in Section 5.3.

A.3 CROSS-BLOCK COUPLING BOUND

Proposition 2 (Alignment with coupled blocks). *Let $K \succ 0$, $D = \mathrm{blkdiag}(K_{11}, \ldots, K_{BB})$, and $E = D^{-1/2}(K - D)D^{-1/2}$. If $\nu = \|E\|_2 < 1$ and $aD \preceq G \preceq bD$ for $0 < a \le b$, then*

$$\kappa(G^{-1/2}KG^{-1/2}) \le \frac{b}{a}\frac{1+\nu}{1-\nu}.$$

Proof. Every principal block of $K \succ 0$ is positive definite, so $D \succ 0$. Since E is symmetric and $\|E\|_2 = \nu$,

$$(1-\nu)I \preceq I + E \preceq (1+\nu)I.$$

Congruence by $D^{1/2}$ gives $(1-\nu)D \preceq K \preceq (1+\nu)D$. Combining this with the metric sandwich yields

$$\frac{a}{1+\nu}K \preceq G \preceq \frac{b}{1-\nu}K.$$

Proposition 1 gives the result. □

If within-block alignment satisfies $(1-\epsilon)D \preceq G \preceq (1+\epsilon)D$, the upper bound is $\frac{1+\epsilon}{1-\epsilon}\frac{1+\nu}{1-\nu}$. The first factor quantifies metric mismatch; the second quantifies ignored coupling. Even $G = D$ leaves the coupling factor. The sufficient condition $\nu < 1$ is restrictive and is not asserted for the benchmark networks. This is an analytical extension of the alignment argument, not a newly measured diagnostic.

A.4 STOCHASTIC ERROR: AN EXACT FROZEN-QUADRATIC RECURSION

Let $G \succ 0$ and $K \succ 0$ be fixed, and consider

$$e_{t+1} = e_t - \eta G^{-1}(Ke_t + \xi_t), \qquad \mathbb{E}[\xi_t \mid \mathcal{F}_t] = 0, \quad \mathrm{Cov}(\xi_t \mid \mathcal{F}_t) = \Sigma_t,$$

where $\mathcal{F}_t$ contains the current iterate and the fixed metric. In whitened coordinates, $z_{t+1} = (I - \eta A)z_t - \eta G^{-1/2}\xi_t$. The cross term has conditional expectation zero. Thus

$$\begin{aligned}\mathbb{E}[\|e_{t+1}\|_G^2 \mid \mathcal{F}_t] &= \|(I-\eta A)z_t\|_2^2 + \eta^2 \operatorname{tr}(G^{-1}\Sigma_t) \\ &\le r(\eta)^2 \|e_t\|_G^2 + \eta^2 \operatorname{tr}(G^{-1}\Sigma_t),\end{aligned} \tag{9}$$

where $r(\eta) = \max_j |1 - \eta\lambda_j(A)|$. If $\operatorname{tr}(G^{-1}\Sigma_t) \le \sigma_G^2$ and $r(\eta) < 1$, iteration gives

$$\mathbb{E}\,\|e_t\|_G^2 \le r(\eta)^{2t}\mathbb{E}\,\|e_0\|_G^2 + \frac{\eta^2\sigma_G^2}{1 - r(\eta)^2}\big(1 - r(\eta)^{2t}\big).$$

The additive noise floor does not vanish merely because conditioning improves. A metric estimated using the same noisy gradient can violate the stated conditional-unbiasedness structure; metric-history dynamics require a separate analysis. No such extension is assumed in interpreting the reported experiments.

A.5 WHAT THE GRADIENT-DISAGREEMENT STATISTIC IMPLIES

At a fixed iterate and fixed metric, write $g^{[i]} = \mu + \epsilon_i$, with independent zero-mean noise and $\mathbb{E}\,\|\epsilon_i\|_g^2 \le \sigma^2$. Then

$$\mathbb{E}\left\|g^{[1]} - g^{[2]}\right\|_g^2 = \mathbb{E}\,\|\epsilon_1\|_g^2 + \mathbb{E}\,\|\epsilon_2\|_g^2 \le 2\sigma^2.$$

This statement alone does *not* justify replacing the random denominator of χ by $\|\mu\|_g^2 + \varepsilon$ inside an expectation.

For a precise bound, assume $m = \|\mu\|_g > 0$ and define $\mathcal{E} = \{\|(\epsilon_1 + \epsilon_2)/2\|_g \le m/2\}$. Independence gives $\mathbb{E}\,\|(\epsilon_1 + \epsilon_2)/2\|_g^2 \le \sigma^2/2$. Markov's inequality therefore implies

$$\Pr(\mathcal{E}^c) \le \min\{1, 2\sigma^2/m^2\}.$$

On $\mathcal{E}$, the denominator is at least $m^2/4 + \varepsilon$, hence

$$\mathbb{E}[\chi \, \mathbf{1}_{\mathcal{E}}] \leq \frac{2\sigma^2}{m^2/4 + \varepsilon}.$$

Without a denominator event or stronger distributional assumptions, the general bound is only $\mathbb{E}\chi \leq 2\sigma^2/\varepsilon$. These bounds explain the signal-to-noise interpretation but do not make a small observed χ a certificate of low population variance. Consecutive checks are dependent, so requiring q checks supplies hysteresis, not an automatic q-fold statistical confidence improvement.

A.6 Positive curvature and descent

In one fixed inner-product space, let H be self-adjoint positive definite and let $\langle s, y\rangle > 0$. Using $v^\flat(w) = \langle v, w\rangle$ and $\varrho = 1/\langle s, y\rangle$, the inverse-BFGS update is

$$H_+ = (I - \varrho s \otimes y^\flat) H (I - \varrho y \otimes s^\flat) + \varrho s \otimes s^\flat.$$

For any $v \neq 0$,

$$\langle v, H_+ v\rangle = \langle w, Hw\rangle + \varrho \langle s, v\rangle^2, \quad w = (I - \varrho y \otimes s^\flat)v.$$

If the second term vanishes, $\langle s, v\rangle = 0$ and $w = v \neq 0$, so the first is positive. Otherwise the second is positive. Thus $H_+ \succ 0$. Limited-memory reconstruction from a positive initial scaling and accepted pairs inherits this property in that common space (Nocedal & Wright, 2006).

For a retraction-smooth loss satisfying

$$\mathcal{L}_{\mathrm{tr}}(R_\theta(v)) \leq \mathcal{L}_{\mathrm{tr}}(\theta) + \langle g, v\rangle + \tfrac{1}{2} L_R \|v\|^2,$$

and an operator with $\underline{\mu} I \preceq H \preceq \overline{\mu} I$, substitution of $v = -\eta H g$ gives

$$\mathcal{L}_{\mathrm{tr}}(R_\theta(-\eta H g)) \leq \mathcal{L}_{\mathrm{tr}}(\theta) - \left(\eta \underline{\mu} - \tfrac{1}{2} L_R \eta^2 \overline{\mu}^2\right) \|g\|^2.$$

It is a descent bound when $0 < \eta < 2\underline{\mu}/(L_R \overline{\mu}^2)$ and $g \neq 0$. It concerns a true-gradient direction with those operator bounds. It is not automatically a monotonicity result for momentum or a stochastic mini-batch line search. Positive definiteness under one metric also does not automatically persist after a metric change or arbitrary non-isometric transport. The common-space requirements in Appendix C are therefore essential.

B Descriptor definitions

Let $x_j \in \mathbb{R}^L$ denote the chronological training trajectory of variable j. Let MA_k be a moving average, $p_{j,\ell}$ its normalized periodogram over L_f retained frequency bins, and $\mathcal{S}_j$ a chosen set of dominant seasonal bins. Using $d > 1$ variables, the estimators are

$$\mathrm{snr} = \frac{1}{d} \sum_{j=1}^{d} \frac{\mathrm{Var}(x_j)}{\mathrm{Var}(x_j - \mathrm{MA}_k(x_j)) + \varepsilon}, \tag{10}$$

$$\mathrm{sent} = -\frac{1}{d \log L_f} \sum_{j=1}^{d} \sum_{\ell=1}^{L_f} p_{j,\ell} \log(p_{j,\ell} + \varepsilon), \tag{11}$$

$$\mathrm{seas} = \frac{1}{d} \sum_{j=1}^{d} \frac{\sum_{\ell \in \mathcal{S}_j} p_{j,\ell}}{\sum_{\ell=1}^{L_f} p_{j,\ell} + \varepsilon}, \tag{12}$$

$$\mathrm{dens} = \frac{2}{d(d-1)} \sum_{1 \leq u < v \leq d} \mathbf{1}\{|\mathrm{corr}(x_u, x_v)| > \tau_g\}, \tag{13}$$

$$\mathrm{gns} = \frac{\|h_1 - h_2\|_2^2}{\|(h_1 + h_2)/2\|_2^2 + \varepsilon}, \qquad h_i = \nabla \ell_{\mathcal{B}_i}(\theta_0). \tag{14}$$

Here $k = 25$, $\tau_g = .5$, and the two batches are independent at the same initialization. The SNR name denotes a variance-to-residual-variance proxy, not an identifiable measurement of the true

signal/noise decomposition. The gradient-disagreement descriptor uses Euclidean gradients at initialization; the online statistic χ_t instead uses the evolving metric.

The sixth entry is the task-defined ratio H/T. The descriptor dependencies and alternative encoders are detailed below.

Table 5: Descriptor roles and data dependencies. None of the six default components uses validation or test targets.

Component	Intended information	Dependency
SNR proxy	Variability relative to local residuals	Training trajectory, moving average
Spectral entropy	Diffuse versus concentrated frequency content	Training spectrum
Seasonality	Mass at selected seasonal frequencies	Training spectrum and bin rule
H/T	Horizon relative to available context	Task definition
Graph density	Cross-series coupling	Training correlations
Gradient disagreement	Initial stochastic update reliability	Training batches and θ_0

The first five components are initialization-independent. Matched initialization and mini-batch draws isolate the initial gradient component in optimizer comparisons. The learned, augmented, and residual descriptors are diagnostic alternatives, not additional components of the six-vector. The augmented vector adds trend, kurtosis, lag-1 autocorrelation, and missingness. The oracle uses validation-selected metric targets and is a validation-assisted reference, not a mathematically guaranteed lower bound.

Standardizing a single six-vector across its coordinates is not equivalent to per-feature standardization across tasks. The latter needs a reference population with nonzero per-feature scale. The reference population and initializer-fitting task split determine how this normalization transfers across tasks; they are distinct from the per-task validation loop.

C Geometric operations and control pseudocode

C.1 Momentum phase

Write $g_t = \operatorname{grad}_g \ell_{\mathcal{B}_t}(\theta_t)$ for the mini-batch metric gradient and $\langle \cdot, \cdot \rangle_t$ for the current metric inner product. In Phase I, the reported NAdam-style update combines transported momentum with a scalar second moment per block:

$$
\begin{aligned}
m_t^{(b)} &= \beta_1 \mathcal{T}_{t-1\to t} m_{t-1}^{(b)} + (1-\beta_1) g_t^{(b)}, \\
v_t^{(b)} &= \beta_2 v_{t-1}^{(b)} + (1-\beta_2) \left\| g_t^{(b)} \right\|_t^2 / n_b, \\
u_t^{(b)} &= \frac{(1-\beta_1) g_t^{(b)} + \beta_1 m_t^{(b)} / (1-\beta_1^t)}{\sqrt{v_t^{(b)} / (1-\beta_2^t)} + \varepsilon}, \qquad \theta_{t+1} = R_{\theta_t}(-\eta_t u_t).
\end{aligned} \tag{15}
$$

This is a specified momentum variant inspired by NAdam and product-manifold adaptation (Dozat, 2016; Bécigneul & Ganea, 2019). Momentum need not be an SPD linear transform of the current gradient, so a preconditioned-gradient descent theorem does not automatically apply to every Phase-I step.

C.2 The gradient is a metric solve on the tangent space

A Riemannian gradient is characterized by $g_\theta(\operatorname{grad}_g L, v) = dL_\theta[v]$ for every tangent v. For ordinary vectors, this gives the solve in Eq. (2). For an embedded constrained block with ambient SPD

Algorithm 1 ROAM: task statistics to metric-controlled training

Require: Training/validation data, backbone blocks, initializer, budgets
Compute c from training data and paired gradients at θ_0.
Set $\gamma^{(0)} = \mathcal{P}(W\widetilde{c} + b)$; initialize four individuals.
Initialize each individual's moments, pair memory, and Phase I.
while the prescribed budget and stopping conditions permit **do**
    for each individual **do**
        Form $G_{\gamma,\theta}$ and solve for g_t by Eq. (2).
        Check χ_t, ρ_t in the same metric/tangent space.
        Switch phase after q consecutive successes or failures.
        Take the Phase-I update or a safeguarded L-BFGS step.
        Maintain consistent curvature memory when geometry changes.
    end for
    At the outer interval, rank using validation/resource utility.
    Copy elites, perturb feasible controls, and account for all work.
end while
Evaluate the validation-selected checkpoint on test.

operator A_b and Euclidean tangent projector P_θ, the induced tangent operator is

$$G_{T,\theta} = (P_\theta A_b P_\theta)|_{T_\theta \mathcal{M}_b}.$$

It is positive definite on the tangent space, and the correct gradient solves

$$G_{T,\theta} g_b = P_\theta \nabla_b L, \qquad g_b \in T_\theta \mathcal{M}_b.$$

In general $A_b^{-1} P_\theta \nabla_b L$ need not be tangent. Thus projection followed by inversion is valid only when the inverse denotes the restricted tangent operator, or when the ambient operator preserves the tangent space. Under the scalar-weighted induced Euclidean metric $A_b = \alpha_b I$, the expression simplifies to $\alpha_b^{-1} P_\theta \nabla_b L$.

C.3 Ordinary, Stiefel, and SPD parameter coordinates

For an ordinary block, $R_\theta(v) = \theta + v$ and coordinate vector transport is the identity. When the metric changes with state, this coordinate transport need not be an isometry of the changing metrics.

For $X \in \mathrm{St}(p, q) = \{X : X^\top X = I_q\}$, $p \geq q$, the tangent space consists of V with $X^\top V + V^\top X = 0$. The induced-Euclidean projector and QR retraction are

$$P_X(Z) = Z - X \,\mathrm{sym}(X^\top Z), \qquad R_X(V) = \mathrm{qf}(X + V),$$

where the QR diagonal sign convention is chosen so $R_X(0) = X$ locally. A projection transport into the new tangent space or a differentiated-retraction transport supplies a valid tangent vector, but need not preserve the old inner product. An orthogonality constraint on a previously unconstrained matrix changes the feasible model family, whereas changing only its optimization metric leaves that family unchanged.

For a positive-definite parameter P, log-Euclidean coordinates write $S = \log P \in \mathrm{Sym}(m)$ and $P = \exp S$ (Arsigny et al., 2006). A symmetric *log-coordinate* step V updates

$$S_+ = S + V, \qquad P_+ = \exp(S + V).$$

The gradient of $L(\exp S)$ must include the differential of the exponential. If $\Xi \in T_P \mathbb{S}_{++}$ is instead a tangent vector in the original P coordinates, the corresponding retraction is

$$R_P(\Xi) = \exp\big(\log P + D \log_P[\Xi]\big).$$

Indeed $D \exp_{\log P} \circ D \log_P$ is the identity. The shorthand $\exp(\log P + \Xi)$ is therefore interpreted only when Ξ denotes a log-coordinate displacement; it is not a valid general retraction formula for a raw P-coordinate tangent.

C.4 Curvature pairs, transport, and changing controls

One consistent destination-space convention for a step $\eta_j p_j$ is

$$s_j = \mathcal{T}_{j\to j+1}(\eta_j p_j), \qquad y_j = g_{j+1} - \mathcal{T}_{j\to j+1} g_j \quad \text{in } T_{\theta_{j+1}}\mathcal{M}.$$

Both gradients refer to the same objective and compatible metric state in this convention. Using different mini-batches mixes curvature with sampling variation; whether batches are shared or refreshed is a separate implementation choice. At a later iterate, stored pairs must be represented in the current tangent space before a two-loop recursion is applied. Curvature must be checked under the inner product used by that recursion. Under non-isometric transport, rechecking positive curvature and rebuilding from a positive initial scaling supplies the common-space BFGS guarantee; alternatively, an isometric transport can preserve it. Reusing unchecked pairs across an arbitrary metric change does not.

When controls or curvature estimates change, a valid implementation must reset, recompute, or consistently transform affected memory. These alternatives characterize the state consistency needed for the fixed-space positive-definiteness result; that result alone does not select a numerical transport policy.

C.5 Control flow and data separation

Algorithm 1 combines per-mini-batch training updates, consecutive-check phase decisions, and slower population evaluations. Descriptors and initial gradient pairs use training data at the shared initialization. Each individual's metric solve feeds the current phase, while the common-space curvature rules govern whether quasi-Newton information remains usable. At an outer evaluation, validation loss and resource utility determine elite copying and feasible-control perturbations. The metric and optimizer state evolve during training; test targets enter only the final evaluation of the selected checkpoint. This separation applies independently of whether the configuration uses ordinary coordinates alone or includes an eligible parameter constraint.

D Experimental specification and reproducibility

This appendix collects the reported numerical configuration, implementation requirements, and evaluation scopes. It complements the metric definition and pseudocode with constants, computational costs, and the distinction between ordinary-weight metrics and optional parameter constraints.

D.1 Specified constants and selection rules

Table 6: Optimizer constants and selection rules.

Quantity	Reported value or rule
β_1, β_2	$0.9, 0.999$
ε, δ	$10^{-8}, 10^{-6}$
L-BFGS memory M	10 accepted pairs
c_{curv}, q	10^{-4}; three consecutive checks
Population and fractions	$N_{\mathrm{pop}} = 4$; elite/replacement $0.25/0.25$
Initial T_{eval}	200 inner steps
Descriptor defaults	Moving-average window 25; graph threshold .5
Shared learning-rate grid	$\{10^{-4}, 3\times10^{-4}, 10^{-3}\}$
Shared weight-decay grid	$\{0, 10^{-4}, 10^{-3}\}$
Model/control selection	Validation only; no test-selected tuning
Seeds	10 for main results and the reported four-setting extension
Environment	Two NVIDIA RTX 4090 GPUs; Ubuntu 22.04; Python 3.10; PyTorch 2.1.0; CUDA 12.1

Chronological train/validation/test splits are reported as following the benchmark convention (Zhou et al., 2021; Wu et al., 2021). Normalization statistics are fit on the training split and applied unchanged to validation/test.

The forecasting evaluations are on normalized targets. A shared learning-rate grid does not by itself prove adequate tuning of methods with different recommended scales or search spaces.

D.2 Block partition and operator storage

The grouping rule fixes a partition within a comparison. The number of groups B depends on the instantiated architecture; shared parameters belong to a single group so that the partition does not duplicate their optimization state.

The expression $\widehat{F}_b = J_b^\top J_b/|\mathcal{B}|$ defines a PSD target proxy. It does not establish whether the stored implementation is diagonal, Kronecker-factored, low-rank, or matrix-free. If J_b is a module-output rather than prediction-residual Jacobian, it is not necessarily the corresponding exact loss Gauss–Newton block.

As algebraic reference points, a diagonal proxy needs $O(n_b)$ storage and solve work. A dense proxy needs $O(n_b^2)$ storage and, with a fresh Cholesky factorization, $O(n_b^3)$ factorization work. For a diagonal-plus-rank-r_b proxy, Woodbury-type solves can use $O(n_b r_b + r_b^2)$ storage and $O(n_b r_b^2 + r_b^3)$ preparation, excluding Jacobian construction. These are alternative implementation costs, not measured properties of the reported ROAM code. The two-loop L-BFGS recursion stores $O(Mn)$ quantities and uses $O(Mn)$ arithmetic in fixed coordinates, plus any transport/metric-solve cost. A four-member population multiplies concurrently resident state when run fully in parallel, or accumulates serial work when run sequentially.

D.3 Configuration dependencies

The metric family, its numerical realization, and the slow controller are three distinct parts of a configuration. Descriptor standardization and the affine initializer determine the initial controls; storage approximation and solve accuracy determine the realized metric direction; transport and memory policies determine which quasi-Newton information remains compatible with that direction. The common-objective condition in Eq. (6) keeps population ranking comparable, while the state-consistency conditions in Appendix C govern curvature-memory validity.

D.4 SPD activation: one consistent interpretation

Table 7: Metric and constraint contributions. The three-setting decomposition and two-model SPD subset (ModernTCN and DLinear) are separate comparisons on distinct populations.

Comparison	Mean MSE
Three-setting decomposition: identity metric	.324
Same decomposition: task metric, ordinary blocks	.313
Same decomposition: task metric plus eligible Stiefel	.312
Separate two-model SPD-eligible subset: SPD off	.324
Same eligible subset: SPD on	.310

The method distinguishes three objects: (i) the SPD *metric* Q_b on an ordinary parameter vector; (ii) a Stiefel *parameter constraint*; and (iii) an SPD *parameter constraint* on a trainable covariance-like matrix. Only (iii) requires SPD-manifold activation. An empirical Fisher matrix or gradient covariance used by the optimizer is not, by itself, an SPD parameter of the forecasting model.

The SPD activation comparison concerns two evaluated configurations, ModernTCN and DLinear, with a matched subset change .324 to .310. This identifies the participating backbones, not an intrinsic SPD parameter in their standard architectures. Table 7 reports this conditional result separately; it does not assign SPD activation to the iTransformer, PatchTST, or TimeMixer runs.

The activation rule is configuration-dependent: if a configuration contains no eligible positive-definite trainable parameter, SPD-manifold operations are disabled; an SPD optimizer metric may still be active. Introducing a positive-definite parameter into a backbone is a parameterization/constraint intervention, distinct from changing the metric on its existing ordinary weights. The

two-model aggregate is a separate on/off comparison, not a per-backbone effect estimate or an activation flag for every benchmark run. The .324-to-.310 subset effect cannot be ranked against the .313-to-.312 Stiefel effect without matched populations. Neither its magnitude nor its provenance supports a blanket statement that SPD is the primary source of the broad forecasting gains.

E FULL FORECASTING RESULTS AND AGGREGATION

The five tables below give all 35 dataset–backbone results. Each result averages the four horizons $\{96, 192, 336, 720\}$ and reports mean $\pm$ standard deviation across 10 seeds. These are not horizon-specific tables; individual horizon results cannot be reconstructed from their averages.

For Table 1, the macro mean is the arithmetic mean of the seven reported dataset means for each backbone. The relative reduction is computed as

$$100\left(1-\frac{\frac{1}{7}\sum_d \mathrm{MSE}_{d,\mathrm{ROAM}}}{\frac{1}{7}\sum_d \mathrm{MSE}_{d,\mathrm{base}}}\right),$$

using the dataset cells before rounding the final displayed macro means. It differs from averaging the seven relative reductions and from taking a ratio of already rounded macro means. The per-dataset summary in Section 5 instead averages the five paired backbone-relative reductions. All derived values inherit the three-decimal rounding of the reported observations.

A standard deviation of a macro average cannot in general be obtained by averaging constituent standard deviations, because seed-wise cross-setting covariance is needed. We therefore report the supplied per-cell deviations without a macro-average confidence interval. No paired significance test is claimed from marginal means and deviations alone.

Table 8: iTransformer: full results, averaged over four horizons. Entries are mean $\pm$ seed standard deviation; B denotes the standard recipe and R denotes ROAM.

Dataset	B MSE	R MSE	B MAE	R MAE
ETTh1	0.454 ± 0.010	0.419 ± 0.008	0.450 ± 0.009	0.424 ± 0.007
ETTh2	0.384 ± 0.009	0.362 ± 0.007	0.407 ± 0.008	0.392 ± 0.006
ETTm1	0.408 ± 0.008	0.379 ± 0.006	0.412 ± 0.007	0.390 ± 0.006
ETTm2	0.292 ± 0.006	0.272 ± 0.005	0.336 ± 0.005	0.319 ± 0.004
Weather	0.261 ± 0.005	0.245 ± 0.004	0.281 ± 0.004	0.266 ± 0.004
ECL	0.176 ± 0.004	0.167 ± 0.003	0.267 ± 0.004	0.259 ± 0.003
Exchange	0.364 ± 0.012	0.349 ± 0.010	0.407 ± 0.010	0.396 ± 0.008

Table 9: PatchTST: full results, averaged over four horizons. Entries are mean $\pm$ seed standard deviation; B denotes the standard recipe and R denotes ROAM.

Dataset	B MSE	R MSE	B MAE	R MAE
ETTh1	0.416 ± 0.009	0.389 ± 0.007	0.430 ± 0.008	0.409 ± 0.007
ETTh2	0.331 ± 0.008	0.315 ± 0.006	0.379 ± 0.007	0.364 ± 0.006
ETTm1	0.352 ± 0.007	0.330 ± 0.006	0.382 ± 0.006	0.365 ± 0.005
ETTm2	0.257 ± 0.005	0.244 ± 0.004	0.314 ± 0.005	0.301 ± 0.004
Weather	0.228 ± 0.004	0.219 ± 0.003	0.264 ± 0.004	0.255 ± 0.003
ECL	0.163 ± 0.003	0.156 ± 0.003	0.254 ± 0.003	0.246 ± 0.003
Exchange	0.386 ± 0.013	0.378 ± 0.010	0.419 ± 0.010	0.411 ± 0.009

Table 10: TimeMixer: full results, averaged over four horizons. Entries are mean ± seed standard deviation; B denotes the standard recipe and R denotes ROAM.

Dataset	B MSE	R MSE	B MAE	R MAE
ETTh1	0.441 ± 0.011	0.413 ± 0.008	0.435 ± 0.009	0.416 ± 0.007
ETTh2	0.392 ± 0.010	0.369 ± 0.007	0.410 ± 0.008	0.393 ± 0.007
ETTm1	0.388 ± 0.008	0.362 ± 0.006	0.399 ± 0.007	0.381 ± 0.006
ETTm2	0.278 ± 0.006	0.262 ± 0.005	0.326 ± 0.005	0.313 ± 0.004
Weather	0.245 ± 0.005	0.232 ± 0.004	0.276 ± 0.004	0.263 ± 0.003
ECL	0.182 ± 0.004	0.174 ± 0.003	0.272 ± 0.004	0.264 ± 0.003
Exchange	0.397 ± 0.013	0.382 ± 0.010	0.423 ± 0.011	0.409 ± 0.009

Table 11: ModernTCN: full results, averaged over four horizons. Entries are mean ± seed standard deviation; B denotes the standard recipe and R denotes ROAM.

Dataset	B MSE	R MSE	B MAE	R MAE
ETTh1	0.428 ± 0.010	0.401 ± 0.008	0.437 ± 0.009	0.419 ± 0.007
ETTh2	0.365 ± 0.009	0.348 ± 0.007	0.393 ± 0.008	0.381 ± 0.006
ETTm1	0.366 ± 0.007	0.347 ± 0.006	0.388 ± 0.006	0.373 ± 0.005
ETTm2	0.266 ± 0.005	0.251 ± 0.004	0.320 ± 0.005	0.308 ± 0.004
Weather	0.236 ± 0.004	0.225 ± 0.003	0.270 ± 0.004	0.259 ± 0.003
ECL	0.170 ± 0.004	0.163 ± 0.003	0.262 ± 0.003	0.255 ± 0.003
Exchange	0.372 ± 0.012	0.358 ± 0.009	0.412 ± 0.010	0.401 ± 0.008

Table 12: DLinear: full results, averaged over four horizons. Entries are mean ± seed standard deviation; B denotes the standard recipe and R denotes ROAM.

Dataset	B MSE	R MSE	B MAE	R MAE
ETTh1	0.433 ± 0.012	0.403 ± 0.009	0.445 ± 0.010	0.421 ± 0.008
ETTh2	0.469 ± 0.014	0.421 ± 0.010	0.463 ± 0.011	0.434 ± 0.009
ETTm1	0.359 ± 0.008	0.341 ± 0.006	0.381 ± 0.007	0.366 ± 0.006
ETTm2	0.283 ± 0.007	0.258 ± 0.005	0.345 ± 0.006	0.321 ± 0.005
Weather	0.247 ± 0.006	0.233 ± 0.004	0.300 ± 0.005	0.281 ± 0.004
ECL	0.167 ± 0.004	0.160 ± 0.003	0.264 ± 0.004	0.257 ± 0.003
Exchange	0.381 ± 0.013	0.352 ± 0.010	0.420 ± 0.011	0.403 ± 0.009

F Detailed mechanism and descriptor ablations

Tables 13 and 14 give the setting-level MSE and MAE for the one-factor ablations. The final column of Table 13 recomputes the mean from those three displayed cells. The corresponding full mean is $.935/3 = .311\overline{6}$, displayed as .312. Using the same unrounded mean denominator, removing the metric gives $100(.324/(.935/3) - 1) = 4.0\%$ after rounding. Removing switching gives 2.5%, removing the outer loop 2.1%, fixed identity plus PBT 4.7%, and random descriptors 5.3%. All percentages use this consistent aggregation convention.

F.1 Factorial marginal effects

Let $f(m, s, o)$ be the reported mean MSE in Table 3. The average metric effect is

$$\Delta_M = \frac{1}{4} \sum_{s,o \in \{0,1\}} \big[f(0, s, o) - f(1, s, o) \big]$$
$$= \tfrac{1}{4}(.010 + .011 + .008 + .012) = .01025.$$

Table 13: Setting-level MSE ablations. A: ETTh1-192; B: ETTm1-336; C: Weather-192. Aggregates are arithmetic means of the displayed cells.

Configuration	A	B	C	Mean
Full ROAM	.362	.341	.232	.312
Without task metric	.376	.355	.241	.324
Without switching	.371	.350	.237	.319
Without outer loop	.369	.348	.238	.318
Fixed identity metric + PBT	.379	.357	.243	.326
Random descriptor metric	.381	.360	.244	.328
Learned descriptor encoder	.364	.343	.233	.313
Validation-assisted oracle	.360	.339	.231	.310
Without SNR	.366	.344	.234	.315
Without spectral entropy	.367	.345	.235	.316
Without graph density	.369	.346	.236	.317
Without gradient disagreement	.371	.349	.238	.319

Table 14: MAE accompanying the MSE observations in Table 13. Setting identifiers are the same.

Configuration	A	B	C
Full ROAM	.394	.364	.263
Without task metric	.406	.376	.271
Without switching	.401	.371	.268
Without outer loop	.399	.370	.269
Fixed identity metric + PBT	.409	.378	.273
Random descriptor metric	.411	.381	.274
Learned descriptor encoder	.395	.365	.264
Validation-assisted oracle	.392	.362	.261
Without SNR	.397	.367	.265
Without spectral entropy	.398	.368	.266
Without graph density	.400	.369	.267
Without gradient disagreement	.402	.371	.268

Likewise $\Delta_S = (.004 + .005 + .003 + .007)/4 = .00475$ and $\Delta_O = (.006 + .004 + .005 + .006)/4 = .00525$. The displayed effects round to .0103, .0048, and .0053. All on/off contrasts are positive, but neither significance nor a population-level interaction is established by rounded means alone.

The O-only factorial cell is .327, whereas the separately evaluated fixed-identity-plus-PBT control rounds to .326. The factorial effects use only the matched cells in Table 3; they do not substitute the latter control for an O-only cell. Likewise, Table 7 separates the three-setting geometry decomposition from the two-model SPD comparison.

F.2 All six removals and richer descriptors

The largest individual removal effects are associated with initial gradient disagreement and graph density: the rounded mean increases by .007 and .005, respectively. Removing either seasonality or SNR increases it by .003. No individual removal matches the .016 deterioration from randomizing the full vector. This pattern is consistent with complementary information across the descriptors, including both a model-dependent initialization statistic and data-derived temporal or cross-series statistics. It does not require every descriptor to contribute equally on every task.

Estimator changes are modest within the tested range: moving-average windows 12/25/48 yield .313/.312/.313, graph thresholds .3/.5/.7 yield .313/.312/.314, and replacing the periodogram with Welch gives .313. Across 10 initializations, the reported gradient-noise descriptor has 4.6% coefficient of variation and downstream MSE standard deviation .003. Bootstrap descriptor cosine similarity is $.97 \pm .01$. These tests support local robustness, not universal descriptor sufficiency or invariance to distribution shift.

Table 15: Descriptor comparisons over ETTh1-192, ETTm1-336, and Weather-192. Mean MSE for descriptor replacements and leave-one-out variants. The oracle is a validation-assisted diagnostic.

Descriptor	Mean MSE
Hand-crafted six	.312
Learned encoder	.313
Validation-assisted oracle diagnostic	.310
Random six-vector	.328
Augmented ten-vector	.311
Hand-crafted plus learned residual	.310
Without SNR	.315
Without spectral entropy	.316
Without seasonality	.315
Without horizon ratio H/T	.316
Without graph density	.317
Without initial gradient disagreement	.319

Table 15 collects the replacement and leave-one-out comparisons.

All six removals worsen the reported mean, and randomization has a larger effect than an individual removal. This supports meaningful task information in the tested descriptor. The comparison measures average predictive utility over the three settings, rather than a universal ranking of descriptor importance. The richer-encoder results are diagnostic accuracy comparisons, not parameter- and compute-matched superiority tests.

G SENSITIVITY AND DYNAMICS DETAILS

Table 16: Reported optimizer sensitivity. Relative epoch time is normalized to the AdamW reference. Values are ordered to match the tested choices within each row.

Choice	Tested values	Mean MSE	Epoch time
L-BFGS memory M	$5/10/20$	$.315/.312/.313$	$1.15/1.19/1.25$
Consecutive checks q	$2/3/5$	$.314/.312/.314$	$1.18/1.19/1.18$
Threshold multiplier	$.75/1/1.25$	$.314/.312/.315$	$1.18/1.19/1.19$
Population size	$2/4/6$	$.316/.312/.311$	$1.10/1.19/1.31$

The reported defaults are best among the tested M, q, and threshold choices. Population size 6 achieves the lowest mean MSE but has higher epoch time than size 4. The default population is therefore an accuracy–resource trade-off, not the unique accuracy optimum.

The base-learning-rate sensitivity reports relative MSE improvements 5.1%, 5.9%, and 5.4% at $0.5\times$, $1\times$, and $2\times$ the base rate. These numbers support persistence of a gain across the tested scales; they do not identify the optimal learning rate for every competitor or task.

Table 17: Reported descriptor-estimation checks. The initialization and bootstrap rows measure different types of variability.

Change	Result
Moving-average window $k = 12/25/48$	Mean MSE $.313/.312/.313$
Correlation threshold $\tau_g = .3/.5/.7$	Mean MSE $.313/.312/.314$
Welch instead of periodogram	Mean MSE .313
Gradient descriptor over 10 initializations	CV 4.6%; downstream MSE s.d. .003
Bootstrap training windows	Descriptor cosine similarity $.97 \pm .01$

For initialization sensitivity, coefficient of variation means standard deviation divided by the mean of the descriptor component, expressed as a percentage. The bootstrap cosine similarity instead measures stability of the full descriptor vector under resampling. Its interpretation depends on temporal dependence in the resampling units, so it is a descriptive stability measure rather than a distribution-free guarantee.

Table 4 gives the four-setting dynamics summary. Its first stable switch step is a mean and seed standard deviation; plateau times and conditioning reductions are supplied as aggregates without their own uncertainty. The conditioning diagnostic is described as normalized effective conditioning. These diagnostic reductions are empirical proxy summaries, not measurements of the exact $\kappa(K)$ or $\kappa(A)$ in the propositions. The aggregate switch statistics describe timing variability but do not determine a full trajectory distribution or phase-reversion rate.

H Optimizer costs and budget accounting

Table 18: ETTh1-96/iTransformer optimizer tournament. Every row reports 28 validation calls. The GPU-hour column gives per-run measured consumption rather than a stopping-budget cap.

Method	MSE	MAE	Plateau min	GPU h/run
AdamW + tuned schedule	.387	.405	71	1.18
Lion + tuned schedule	.379	.398	68	1.13
SAM + tuned radius	.371	.392	74	1.22
Sophia-G + tuned parameters	.366	.389	61	1.17
PBT-only	.370	.392	66	1.18
Riemannian-only	.365	.388	59	1.19
NAdam–L-BFGS, no task metric	.363	.386	57	1.18
Random descriptor metric	.372	.395	63	1.17
ROAM	.358	.383	46	1.18

Table 19: Per-epoch and memory overhead on ETTh1-96/iTransformer. Peak GPU memory is reported in GB. These values do not by themselves provide a population-wide compute ledger.

Method	Relative epoch time	Plateau (min)	Peak memory (GB)
AdamW	1.00	71	16.1
Lion	.95	68	16.0
SAM	1.52	74	16.8
Sophia-G	1.12	61	16.6
ROAM	1.19	46	17.4

ROAM's relative epoch overhead is 19%, memory increase is $100(17.4/16.1 - 1) = 8.1\%$, and plateau reduction is $100(1 - 46/71) = 35.2\%$. Neither an epoch-time ratio nor a plateau duration determines total GPU-hours, particularly with parallel population members. The final experimental suite is reported to use approximately 500 GPU-hours excluding exploratory pilots; this aggregate complements, rather than replaces, individual run accounting.

Population-wide compute includes all individuals, replacements, unsuccessful runs, curvature/descriptor work, and line-search evaluations. With parallel individuals, elapsed wall-clock time and summed device-hours measure different resources; similarly, one population evaluation round can involve several model-level validation calls. The reported per-run costs do not disaggregate all of these contributions. Accordingly, the results establish an accuracy–overhead trade-off under the reported budget protocol, rather than an identical end-to-end resource cost for every method.

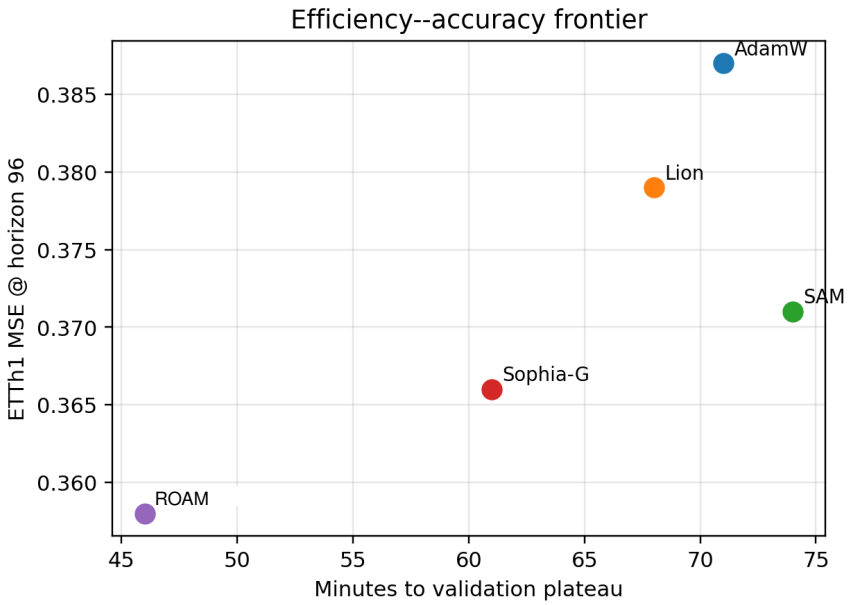

Figure 5: Accuracy–plateau-time frontier on ETTh1-96. ROAM reaches a lower test MSE and an earlier validation plateau than the displayed AdamW, Lion, SAM, and Sophia-G runs. The expanded comparison is in Figure 3; Tables 18 and 19 distinguish epoch time, plateau time, and GPU-hour accounting.

I SUPPLEMENTARY TRANSFER RESULTS

Table 20: Source–target transfer summary. Cells are MSE/MAE. Relative gains are computed from the displayed MSE values.

Source → target	Standard recipe	ROAM	MSE gain
ETTh1 → ETTh2	.421/.438	.392/.414	6.9%
ETTm1 → Weather	.286/.332	.271/.318	5.2%
Weather → ECL	.201/.284	.189/.271	6.0%
ECL → Exchange	.462/.486	.437/.463	5.4%
Mean over 12 transfers	.342/.385	.322/.366	5.8%

The table gives four representative source–target pairs and a 12-transfer mean. These aggregate results show lower target-task errors, but do not isolate transfer of the predictor from transfer of the optimizer initializer or metric controls. Target-data adaptation is also distinct from zero-shot prediction: training a predictor on target data with transferred controls still adapts that predictor. The table is therefore supplementary evidence of source–target performance, not a zero-shot forecasting result or a component-specific transfer attribution.

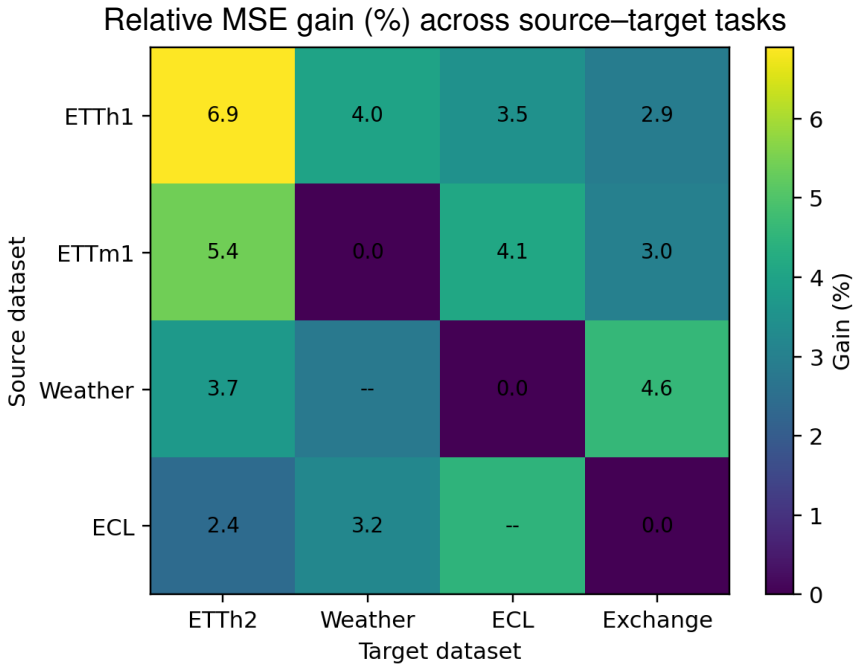

Figure 6: Cell-wise source–target diagnostic. Rows specify source datasets and columns specify targets; entries are relative MSE gains (%). Dashes denote unreported cells. This heatmap is a separate diagnostic from the aggregate MSE/MAE summary in Table 20; the table's 12-transfer mean is not an average of these displayed cells. The diagnostic concerns source–target transfer and does not by itself identify whether the transferred object is a predictor or an optimizer control.

J ADDITIONAL VISUAL DIAGNOSTICS

J.1 FORECAST BEHAVIOR

Figure 7 complements the aggregate benchmark with two representative forecast traces. It displays the shape and timing of predictions relative to the observed history and future target; it is a qualitative view, not a replacement for multi-seed MSE/MAE.

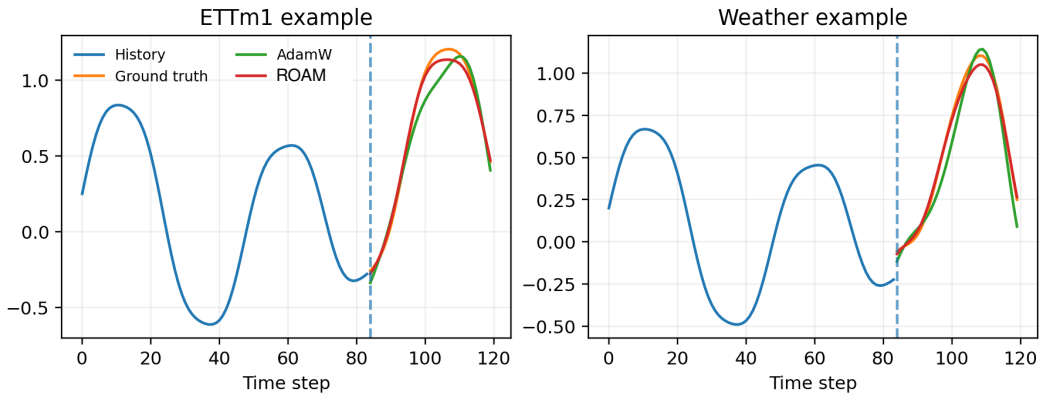

Figure 7: Representative ETTm1 and Weather forecasts. The dashed vertical line marks the transition from observed history to the forecast interval. Ground truth, AdamW, and ROAM are shown on the same axes; differences around turning points illustrate the aggregate comparison without implying uniform superiority at every time step.

J.2 DESCRIPTOR ABLATION VISUALIZATION

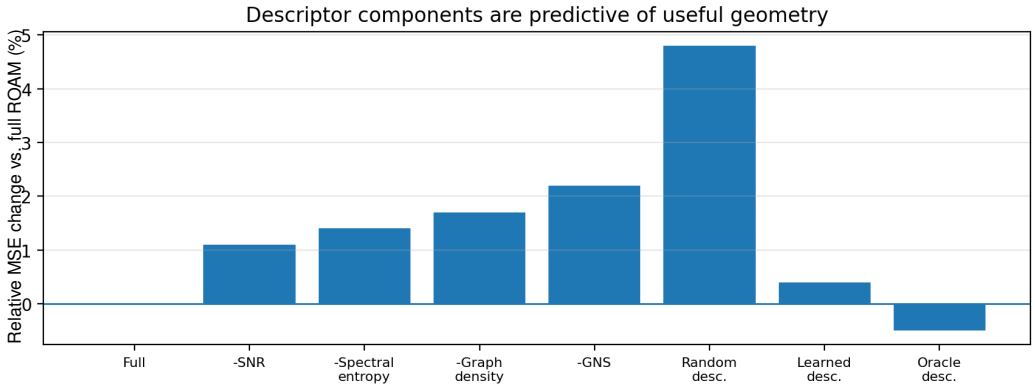

Figure 8: Descriptor ablation diagnostic. Positive bars denote increased MSE relative to the full descriptor, and the oracle is a validation-assisted reference. The plotted percentages are the reported relative-change summaries; Tables 15 and 13 provide mean and setting-level comparisons.

J.3 OPTIMIZATION ANISOTROPY

Figure 9 complements the training trajectories in Figure 4 with normalized conditioning, gradient-noise, sharpness, and stiffness diagnostics.

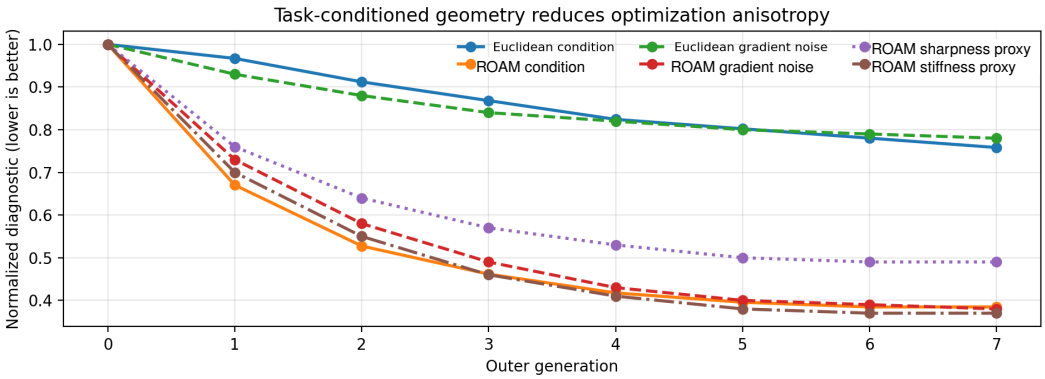

Figure 9: Normalized optimization diagnostics across outer generations (lower is better). Conditioning and gradient-noise curves compare Euclidean and task-conditioned metrics; sharpness and stiffness traces track the task-conditioned run. All displayed diagnostics are normalized to their initial values and are distinct from full-Hessian guarantees.

K Related Work

K.1 Forecasting backbones have advanced rapidly, but optimization is still treated as generic

Modern forecasting has been shaped by a sequence of strong architectural advances. Earlier neural forecasting systems emphasized explicit temporal structure through recurrence, convolution, or basis expansion. LSTNet combined convolutional and recurrent modules to capture local and long-term dynamics (Lai et al., 2018), while N-BEATS showed that deep residual basis expansions can provide a powerful and interpretable alternative for forecasting (Oreshkin et al., 2020). These models established an important lesson: forecasting gains can come from structural priors rather than brute-force sequence modeling.

The next major wave was driven by Transformers. Informer introduced sparse attention for efficient long-range forecasting (Zhou et al., 2021); Autoformer replaced standard attention with decomposition and auto-correlation (Wu et al., 2021); and FEDformer pushed this line into the frequency domain to better model global temporal structure (Zhou et al., 2022). Crossformer then highlighted cross-dimension dependencies in multivariate forecasting (Zhang & Yan, 2023). PatchTST simplified the forecasting pipeline by using patch tokenization and channel independence, and demonstrated that this combination yields a surprisingly strong long-term forecasting baseline (Nie et al., 2023). iTransformer argued that the conventional temporal-token perspective itself is suboptimal and proposed an inverted variate-centric view that is especially effective for multivariate forecasting (Liu et al., 2024b). In parallel, MICN and TimesNet showed that local-global context modeling and temporal 2D variation are powerful alternatives to standard attention (Wang et al., 2023; Wu et al., 2023).

A key correction to this architecture race came from DLinear, which showed that simple linear baselines can outperform many elaborate Transformer forecasters on standard benchmarks (Zeng et al., 2023). This result forced the community to re-examine the sources of apparent progress. TimeMixer and ModernTCN continued this rebalancing by demonstrating that decomposable multi-scale mixing and modern pure-convolution designs can remain highly competitive (Wang et al., 2024; Luo & Wang, 2024). Koopa added another perspective by linking time-series forecasting to Koopman-style operator learning and non-stationary decomposition (Liu et al., 2023). The broader message of these papers is that forecasting performance is not monopolized by any single model family.

Recent foundation-model efforts have pushed the field even further. Time-LLM reprograms large language models for time-series forecasting (Jin et al., 2024); Timer casts time-series modeling as large-scale generative pretraining (Liu et al., 2024c); TimesFM introduces a decoder-only foundation model with strong zero-shot behavior (Das et al., 2024); MOMENT explores open foundation models for general time-series analysis (Goswami et al., 2024); Chronos develops tokenized probabilistic forecasting in TMLR (Ansari et al., 2024); and Moirai pursues universal forecasting transformers trained across diverse temporal corpora (Woo et al., 2024). These models have expanded the ambition of forecasting from in-domain benchmark fitting to cross-domain temporal intelligence.

However, despite this breadth, the overwhelming majority of forecasting papers still leave the training process itself largely generic. They differ in encoder structure, tokenization, decomposition, or pretraining strategy, but are optimized in essentially the same way: a fixed Euclidean parameterization, a broadly standard optimizer, and a manually selected schedule. This is the first major limitation of the literature from our perspective. Once models are strong enough, performance is not determined only by representational capacity; it is also shaped by whether the training dynamics are matched to task structure. The forecasting literature has not yet developed this point into a systematic framework.

K.2 Forecasting objectives have diversified, but training geometry remains largely fixed

Forecasting has also diversified at the level of objective functions and predictive formulations. Pointwise losses such as MSE and MAE remain dominant, but many works implicitly or explicitly acknowledge their limitations. Temporal Fusion Transformers introduced probabilistic forecasting with quantile outputs and interpretability mechanisms (Lim et al., 2021). Decomposition-based

models such as Autoformer and FEDformer emphasize trend-seasonality structure (Wu et al., 2021; Zhou et al., 2022). Patch-based and mixer-based methods exploit local stationarity and multi-scale composition (Nie et al., 2023; Wang et al., 2024). Foundation models further emphasize zero-shot robustness, calibration, and broad-domain transfer (Das et al., 2024; Ansari et al., 2024; Goswami et al., 2024).

Yet these developments mostly act on the *prediction side* of forecasting rather than on the *optimization side*. Even when models output uncertainty-aware or probabilistic forecasts, training is still typically conducted in a task-agnostic Euclidean parameter space. In practice, optimizers such as Adam and AdamW remain standard defaults (Kingma & Ba, 2015; Loshchilov & Hutter, 2019), while more recent alternatives such as SAM, Lion, and Sophia seek improved generalization or faster convergence through gradient statistics, update rules, or local objectives (Foret et al., 2021; Chen et al., 2023; Liu et al., 2024a). These methods are strong and important, but they remain essentially generic: they do not use explicit forecasting-task descriptors to select local metrics, update phases, and slow-timescale control strategies jointly.

This matters because forecasting datasets are not interchangeable optimization problems. Electricity, weather, and exchange-rate data differ in noise scale, horizon sensitivity, cross-variate structure, and non-stationarity. These differences are not only statistical; they are also *optimization-geometric*. They affect gradient-noise scale, anisotropy, local conditioning, and the reliability of curvature information. A training recipe that works for one regime may be poorly aligned with another. Existing forecasting literature has not made this mismatch explicit enough. Our work treats this as the central missing layer: forecasting requires not only expressive architectures, but also *task-conditioned training geometry*.

K.3 Geometry-aware optimization is mature in theory, but underused in forecasting

The broader optimization literature provides a strong foundation for such a view. Natural gradient methods, originating with Amari, show that steepest descent should be defined relative to information geometry rather than raw coordinates (Amari, 1998). Matrix-manifold optimization later systematized optimization over structured spaces through retractions, vector transport, and Riemannian first- and second-order methods (Absil et al., 2008). These tools are especially useful when parameters have intrinsic structure, such as orthogonality or positive-definiteness. Bonnabel extended stochastic gradient descent to Riemannian manifolds with convergence analysis (Bonnabel, 2013), and Bécigneul and Ganea generalized adaptive methods to products of manifolds (Bécigneul & Ganea, 2019). Collectively, these results establish that geometry influences descent direction, conditioning, invariance, and optimization efficiency.

However, this line of work has only had limited influence on mainstream time-series forecasting. In many applications, manifold-aware optimization is introduced only when a model contains an explicit non-Euclidean constraint, rather than as a general principle for aligning learning dynamics with task structure. Forecasting, by contrast, offers a richer opportunity: the challenge is not merely to satisfy constraints, but to choose a geometry appropriate to the task. To our knowledge, existing forecasting methods rarely formulate geometry as a learned object induced by task statistics such as noise level, spectral complexity, cross-series coupling, or horizon difficulty. In other words, geometric optimization theory exists, but forecasting has not yet fully absorbed its implications.

This gap becomes even more important when one considers phase structure in training. In deep forecasting, early training is often dominated by noisy gradients and unreliable curvature estimates, whereas later training operates in a lower-noise regime where second-order information becomes more useful. Generic optimizers partially adapt to this transition, but they do not explicitly represent it. Our framework makes this structure central by introducing a *phase-aware geometric inner loop* that transitions from a noise-robust first-order regime to a curvature-aware quasi-second-order regime. This is not a cosmetic detail: it is how the geometry becomes operational across training stages.

K.4 Outer-loop adaptation exists, but it usually does not adapt geometry

Population-Based Training (PBT) demonstrated that strong training schedules can be discovered online by evolving hyperparameters together with model parameters (Jaderberg et al., 2017). This insight is especially relevant for forecasting, where horizon length, seasonality, and dataset scale often change the preferred learning-rate schedule, regularization strength, or stopping behavior. More generally, PBT and related adaptive control methods established that training dynamics need not be fixed in advance.

But standard outer-loop adaptation has a major blind spot: it usually searches over schedules in a fixed Euclidean training space. It may adapt learning rate, dropout, weight decay, or augmentation strength, but it does not ask whether the *geometry* of parameter updates should also vary across tasks. From our viewpoint, this is why schedule adaptation alone is not enough. If the underlying metric is mismatched to the task, then even well-chosen schedules can only partially compensate. This is exactly what our own ablation logic is designed to test: Euclidean training plus outer adaptation remains weaker than a framework in which outer adaptation acts on a task-conditioned geometric inner loop.

K.5 Recent bilevel optimization on manifolds opens the door, but the forecasting formulation is still missing

A final relevant stream is recent work on bilevel optimization over Riemannian manifolds. These methods study how slow and fast variables can be optimized jointly when the feasible set or geometry is non-Euclidean (Han et al., 2024; Li & Ma, 2025). This is directly connected to our setting, because our framework explicitly contains a *fast geometric inner loop* and a *slow evolutionary outer loop*. Bilevel optimization therefore provides a conceptual language for unifying geometry choice, optimizer phase control, and resource-aware adaptation. ROAM uses a derivative-free population controller rather than the hypergradient algorithms analyzed in these works.

Yet even here, a forecasting-specific formulation is still missing in the literature. Existing manifold bilevel methods are mathematically important, but they are not designed around the specific heterogeneity of forecasting tasks. They do not ask how spectral structure, cross-variate coupling, horizon difficulty, and transfer requirements should shape geometry. Nor do they connect bilevel adaptation to the empirical regularities of forecasting benchmarks. Our work can thus be seen as the forecasting instantiation of a more general idea: geometry and training controls should be adapted jointly, but the adaptation must be driven by the statistics of temporal tasks rather than by abstract hyperparameter search alone.

K.6 Why our formulation is necessary

Taken together, the literature points to a clear unresolved problem. Forecasting architecture papers explain how to model temporal structure, but usually assume a largely generic training process. Geometry-aware optimization explains why descent depends on metric and curvature, but is rarely specialized to heterogeneous forecasting. Population-based and bilevel methods explain how to adapt training dynamics over time, but usually not how to adapt the geometry itself. Foundation models raise the importance of cross-task robustness, but make the weakness of fixed dataset-specific training even more visible.

This is precisely the gap that motivates ROAM. We need a training framework in which task heterogeneity is represented explicitly, optimization geometry is induced by task statistics, inner-loop dynamics are phase-aware, and outer-loop adaptation is resource-aware and validation-driven. Our model is necessary because no existing line of work fully resolves this combination. It is therefore better understood not as a stitched collection of manifold optimization, adaptive scheduling, and PBT, but as a unified response to a missing principle in forecasting: *task structure should determine optimization geometry*.