Adaptive Hyperparameter Optimization for Time-Series Forecasting Models: A Reinforcement Learning Approach

Tomás Martínez*

Universidad Nacional de Rosario,Rosario,Argentina

Abstract:Deep neural networks have become the dominant backbone for multivariate time-series forecasting, yet their prediction quality remains tightly coupled to a hand-specified hyperparameter configuration. Conventional tuning practices rely on grid search, random sampling, or Bayesian surrogates that treat the forecasting model as a static black box and cannot reuse the structure discovered during earlier evaluations. This study formulates hyperparameter tuning for recurrent and Transformer forecasters as a finite-horizon Markov decision process and solves it with a proximal policy optimization (PPO) agent that proposes mixed discrete-continuous configurations, observes validation performance and search history, and receives a shaped reward built from normalized forecast error and search-efficiency cost. The agent operates on three target architectures, namely LSTM, GRU, and an Informer-style Transformer, and shares a single policy across them. Experiments on ETT-h1, ECL, and Weather benchmarks with horizons of 96, 192, 336, and 720 steps show that the proposed tuner reduces the mean test MAE by 6.7%, 7.5%, and 8.5% relative to Bayesian optimization on the three datasets while consuming 33% fewer evaluated configurations. Ablation experiments isolate the contribution of improvement shaping, history features, the efficiency penalty, and the PPO actor-critic update, and a sensitivity analysis maps the joint response of learning rate and hidden dimension. The method provides a reusable, self-improving tuning loop that adapts its search policy to the target dataset rather than restarting a surrogate from scratch.

Index Terms—Time-series forecasting, hyperparameter optimization, reinforcement learning, proximal policy optimization, neural architecture configuration, adaptive search.

1 Introduction

Multivariate time-series forecasting underpins a wide range of decision systems, from power-grid load management and industrial process monitoring to financial risk measurement and environmental monitoring[1,2]. Over the past decade, deep sequence models have displaced classical statistical forecasters in many of these settings: recurrent networks with long short-term memory (LSTM) and gated recurrent units (GRU) capture long-range temporal dependencies, while attention-based Transformers and their long-context variants such as Informer model pairwise temporal interactions without recurrence. Empirical surveys of forecasting competitions confirm that model choice alone explains only part of the achievable accuracy, and that the configuration of optimizer, capacity, regularization, and batch conditions frequently changes

reported error by a larger margin than the choice between competing architectures. This distribution-sensitive tuning challenge is not exclusive to time-series forecasting; large-scale geospatial foundation models for remote sensing, such as CrossEarth-SAR, are also designed to tackle domain shifts across diverse SAR observation conditions, yet still rely on static, dataset-specific hyperparameter setups [3].

The forecasting setting adds two complications absent from image classification. First, the training objective is evaluated over multiple horizons simultaneously, so a configuration that excels at 96-step prediction may overfit local momentum and fail at 720 steps; the tuner must weigh horizon-balanced error rather than a single static accuracy[4-6]. Second, channel correlation and seasonal structure differ between datasets, so a configuration tuned on one series frequently degrades on another, which argues against shipping a universal default. Hyperparameter optimization (HPO) is therefore an inseparable component of any competitive forecasting pipeline. Classical practices cover three families: grid search enumerates a factorial combination of levels but scales poorly and wastes evaluations on obviously poor regions; random search samples uniformly and benefits from the low effective dimensionality of many problems[7-10], yet offers no mechanism to concentrate samples around promising regions[11-13]; Bayesian optimization (BO) fits a surrogate (typically a Gaussian process or tree-structured density estimator) over past observations and selects the next point by an acquisition function such as expected improvement, delivering good sample efficiency on small spaces but degrading when the space mixes categorical and continuous variables and when function evaluations are themselves cheap and numerous[14-16]. Successive-halving and Hyperband-style methods add multi-fidelity promotion that terminates poor configurations early, but their race mechanism remains agnostic to the structure of the forecasting objective and discards information across races[17-20].

A shared limitation of these methods is that they are memoryless with respect to the strategy that produced past observations: a Gaussian process surrogate restarts on every new dataset, and random sampling has no internal model of which regions were informative. Reinforcement learning (RL)[21-24], by contrast, learns a behavioral policy that maps observed search context to actions and improves it through repeated interaction. Its early successes in neural architecture search demonstrate that an agent can learn to propose good configurations rather than merely discard bad samples[25-27]. However, RL-based configuration has been studied mainly for image classifiers, whose loss landscape, budget structure, and action space differ substantially from those of long-horizon sequence forecasting: forecasting training is comparatively inexpensive per configuration, validation error is non-stationary across horizons, and the search space mixes continuous learning rates, integer capacities, categorical architectural choices, and dropout regularization[28-30].

The present study targets this gap with an RL-HPO loop dedicated to time-series forecasters. The contributions are as follows. First, the tuning problem is formalized as a Markov decision process in which the state encodes current validation error, aggregated search history[31-33], and remaining budget; the action encodes a mixed discrete-continuous configuration; and the reward couples normalized MAE/RMSE improvement with an explicit efficiency penalty. Second, a PPO actor-critic agent is trained over these episodes[34-37], with a mixed-distribution policy that emits continuous parameters through a Gaussian head and categorical parameters through a softmax head, and a shaped reward that rewards improvement rather than absolute error. Third, a controlled experimental campaign against manual tuning[38-40], grid search, random search, BO, and ASHA

on three public benchmarks quantifies both prediction accuracy and search cost, and ablation and sensitivity analyses identify which components drive the gains.

Three research questions guide the study. (RQ1) Can an RL policy learn to bias configuration proposals toward regions that reduce validation error faster than memoryless surrogates under a matched budget? (RQ2) Which components of the agent — improvement shaping, history context, efficiency penalty, or the PPO estimator — account for the measured gain? (RQ3) How does the learned tuner trade prediction accuracy against search cost across datasets with different channel counts and seasonal structure? Answering RQ1 requires a controlled comparison against mature baselines; RQ2 requires factorial ablation; RQ3 requires reporting both error and consumed evaluations side by side.

The remainder of the paper is organized as follows. Section 2 reviews related work on deep forecasting models, HPO baselines, and RL-driven configuration. Section 3 formalizes the environment, state, action, and reward and derives the PPO update. Section 4 describes datasets, baselines, metrics, and implementation details. Section 5 reports main results, convergence, efficiency, and sensitivity. Section 6 presents ablation experiments, Section 7 discusses limitations and threats to validity, and Section 8 concludes.

2 Related Work

2.1 Deep models for multivariate time-series forecasting

Classical forecasting rests on Box–Jenkins linear stochastic models and exponential-smoothing families, which remain competitive for univariate seasonal series but scale poorly to hundreds of interdependent channels[41-43]. Recurrent neural networks process the input sequence stepwise and introduce gating to suppress vanishing gradients: LSTM adds input, forget, and output gates, and GRU merges them into reset and update gates with fewer parameters. Attention mechanisms compute pairwise similarities between queries and keys and dispense with recurrence; Informer replaces the full self-attention with prob-sparse attention and proposes generative decoding for long sequences, while Autoformer decomposes the series into trend and seasonality inside the network and FEDformer operates in the frequency domain[44-47]. A later critical study argues that many Transformer forecasters gain little over linear channel-independent models and emphasizes that architectural gains are entangled with tuning. TimesNet converts temporal variation into a two-dimensional representation, and LSTNet combines convolutional extraction with recurrent skip connections. Across these families, empirical evidence repeatedly indicates that reported rankings change when training configuration is varied, motivating a dedicated tuning layer rather than a fixed default configuration[48-50]. Recurrent and attention models also interact differently with optimizer configuration: gated networks are comparatively robust to learning-rate scale, whereas attention layers need smaller, carefully annealed rates and explicit regularization to prevent softmax saturation; a tuner that treats all three families identically under-serves at least two of them. This observation motivates a single policy that conditions its action on the target model identity rather than a universal fixed grid.

2.2 Hyperparameter optimization baselines

Grid search is the oldest exhaustive strategy and remains common in small spaces; its cost grows combinatorially with the number of levels. Random search samples each dimension independently and was shown to outperform grid search when few dimensions are influential, because it explores continuous margins more evenly. Bayesian optimization , maintains a probabilistic surrogate and an acquisition function such as expected improvement; Gaussian-process variants scale awkwardly with dataset size, so tree-based estimators,and multi-objective extensions are widely used in practice. Hyperband and BOHB allocate increasingly many resources to configurations that win successive races, and ASHA extends racing asynchronously. General automated-ML frameworks,package these strategies together, and Optuna provides a programmable sampler interface. Two practical weaknesses recur across these methods. First, surrogate-based acquisition treats every new dataset as fresh evidence: the Gaussian-process kernel and its length scales are re-estimated from scratch, so early iterations under uncertainty revisit broad regions that a seasoned searcher would already avoid. Second, multi-fidelity racing terminates weak configurations early but carries no memory of which region produced the promoted survivors, and repeated races on the same dataset discover the same landscape redundantly. These methods are strong baselines but share the assumption that the search distribution should be re-derived from observations alone; none learns a transferable policy that biases proposal distribution from past datasets or past tuning campaigns. The gap is not that classical methods fail, but that they cannot accumulate strategic experience across campaigns.

2.3 Reinforcement learning for configuration and architecture

RL-driven configuration casts the search space as an action space and the objective as a reward. Baker et al.use an agent to layer-by-layer design convolutional networks; ENAS parameter-shares child models and trains a controller with policy gradient; Regularized Evolution evolves architectures under a population; DeepArchitect separates representation from search and reports controller-driven design. Outside architecture search, RL has been used to learn optimizer schedules, curriculum strategies, and data-collection policies. PPO replaces the high-variance REINFORCE estimator with a clipped surrogate that constrains the policy ratio and is robust in continuous and mixed action spaces; its predecessor DQN handles discrete actions by value iteration but struggles with the mixed spaces required here. Despite this lineage, published RL-HPO studies seldom target long-sequence forecasting, where horizons vary, validation error drifts across capacity levels, and the relevant action space includes learning-rate and regularization choices that govern generalization rather than topology. Image-classification controllers operate on discrete layer-building blocks and reward a static accuracy, whereas a forecasting agent must accept mixed continuous factors, reward incremental improvement rather than final accuracy, and respect a cost budget that real training imposes. The present work adapts the PPO loop to this setting, designs a forecasting-specific state and reward, and evaluates it under matched budgets against mature non-RL baselines. The comparison is deliberately held at equal evaluation budgets so that accuracy gains are not bought by simply trying more configurations.

3 Methodology

3.1 Problem statement

Let a multivariate sequence $X_{1:T} \in \mathbb{R}^{T\times C}$ contain C channels observed at integer time steps. Given an input window of length L_{in}, a forecaster f_ψ parameterized by configuration ψ produces a H-step-ahead prediction:

$$\hat{Y}_{t+1:t+H} = f_\psi\left(X_{t-L_{in}+1:t}\right), \qquad \hat{y}_{t,i} = \hat{Y}_{t,i}, \qquad y_{t,i} \in \mathbb{R}. (1)$$

Training minimizes the multistep empirical risk over the training split:

$$\mathcal{L}(\psi) = \frac{1}{N_H}\sum_t \sum_{i=1}^{H} \| y_{t,i} - \hat{y}_{t,i} \|_2^2, (2)$$

where N_H counts valid windows. The four evaluation metrics used throughout are MAE, RMSE, MAPE, and coefficient of determination R^2:

$$MAE = \frac{1}{N}\sum_{n=1}^{N}|y_n - \hat{y}_n|, (3)$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(y_n - \hat{y}_n)^2}, (4)$$

$$MAPE = \frac{100}{N}\sum_{n=1}^{N}\left|\frac{y_n - \hat{y}_n}{y_n}\right|, (5)$$

$$R^2 = 1 - \frac{\sum_{n=1}^{N}(y_n - \hat{y}_n)^2}{\sum_{n=1}^{N}(y_n - \bar{y})^2}. (6)$$

The tuning objective is to find the configuration $\psi^\star$ that minimizes the validation MAE (used as the tuning metric) while keeping training cost bounded; the test split is touched only after tuning terminates. Two design choices follow from this formulation. First, validation MAE, rather than training loss, guides the search: training loss alone can keep decreasing as capacity grows while generalization degrades, so optimizing it would push the agent toward over-parameterized trials. Second, the same configuration must perform across all four horizons; averaging horizon-specific error inside the validation loop prevents a configuration from being selected for its performance at a single horizon, which would produce an artificially narrow optimum that fails at longer leads. The look-back window L_{in} is held fixed across campaigns so that the tuning budget is spent on optimization rather than on input-length selection.

3.2 Markov decision process formulation

Tuning episodes are cast as a finite-horizon Markov decision process (MDP) $\langle \mathcal{S}, \mathcal{A}, P, r, \gamma \rangle$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ the action space, $P(s_{t+1}|s_t, a_t)$ the environment transition induced by training and validating a configuration, $r: \mathcal{S}\times\mathcal{A} \to \mathbb{R}$ the scalar reward, and $\gamma \in [0,1)$ the discount factor:

$$G_t = \sum_{k=0}^{T-t}\gamma^k r_{t+k}, \qquad \max_\pi \mathbb{E}_{\tau\sim\pi}[G_t]. (7)$$

An episode lasts until the evaluation budget is exhausted or a termination criterion is met; each transition corresponds to one proposed configuration evaluated on the validation split.

3.2.1 State space

The state at transition t contains four blocks. The first block reports the most recent validation error pair $\left(e_t^{\mathrm{MAE}}, e_t^{\mathrm{RMSE}}\right)$ together with the running best validation error $e^{\star}$ observed so far. The second block summarizes the search history: the mean and standard deviation of evaluated MAE values, the fraction of evaluations that improved the best metric, and the age (in steps) of the current best record. The third block encodes the target base model (one-hot over LSTM, GRU, Transformer) and the active horizon family. The fourth block encodes the remaining budget: fraction of evaluations left and fraction of wall-clock time left. The state vector is

$$s_t = \left[e_t^{\mathrm{MAE}}, e_t^{\mathrm{RMSE}}, e^{\star},\ \mu_t^{\mathrm{hist}}, \sigma_t^{\mathrm{hist}}, \rho_t^{\mathrm{imp}}, \Delta_t^{\mathrm{age}},\ o_{\mathrm{model}}, o_{\mathrm{horizon}},\ b_t^{\mathrm{eval}}, b_t^{\mathrm{time}}\right] \in \mathbb{R}^{d_s}, \quad (8)$$

with $d_s = 14$ after one-hot encoding. All continuous state entries are min-max normalized using running statistics collected during the first rollout batch; stationary normalization keeps the policy network numerically well behaved across datasets with different error scales.

3.2.2 Action space

The action a_t directly parameterizes the hyperparameter configuration ψ_t through a mixed distribution. Continuous factors (learning rate, dropout, weight decay) are emitted as Gaussian samples transformed to their support; integer and categorical factors (hidden dimension, number of layers, batch size, attention heads) are emitted as categorical samples. Concretely,

$$a_t = \left[\eta_t,\ h_t,\ L_t,\ p_t^{\mathrm{drop}},\ B_t,\ n_t^{\mathrm{head}},\ \lambda_t^{\mathrm{wd}}\right], \quad (9)$$

where η_t is the learning rate on a log scale, h_t the hidden dimension, L_t the number of stacked layers, p_t^{drop} the dropout probability, B_t the batch size, n_t^{head} the number of attention heads (Transformer only), and λ_t^{wd} the weight-decay coefficient. The decoder rounds continuous samples to the search-space levels shown in Table II and rejects infeasible combinations (e.g., attention heads for recurrent models) by masking the corresponding logits before sampling.

3.2.3 Reward function

The reward combines three terms. The primary term rewards normalized improvement in validation MAE relative to the best record so far; the secondary term subtracts a normalized RMSE contribution to discourage MAE/RMSE imbalance; the third term penalizes evaluation cost so that the agent prefers cheaper configurations when error differences are small:

$$r_t = -\alpha\,\widetilde{\mathrm{MAE}}_t - \beta\,\widetilde{\mathrm{RMSE}}_t + \kappa\,\Delta e_t^{+} - \lambda_{\mathrm{eff}}\,c_t, \quad (10)$$

where $\widetilde{\mathrm{MAE}}_t$ and $\widetilde{\mathrm{RMSE}}_t$ are min-max normalized validation errors, $\Delta e_t^{+} = \max\left(0, e^{\star} - e_t^{\mathrm{MAE}}\right)$ is the shaped improvement bonus, and c_t is the normalized training cost (training epochs consumed plus promoted fractional budget). Coefficients are $\alpha = 0.4$, $\beta = 0.2$, $\kappa = 1.0$, and $\lambda_{\mathrm{eff}} = 0.15$ throughout. Normalization uses running minima and maxima observed so far rather than dataset-wide constants, so the reward remains comparable when early trials are unusually poor; once a new extreme appears, the window updates and old observations are rescaled retrospectively within the rollout batch. The improvement bonus Δe_t^{+} is essential: without it the agent receives a signal proportional to absolute error and cannot distinguish a small genuine improvement from noise; shaping converts episode-level sparse success into dense feedback. A non-positive bonus keeps the reward bounded below by the cost term, which prevents episodes dominated by repeated exploration from receiving spuriously high returns and destabilizing the critic.

3.3 PPO agent and interaction loop

The policy $\pi_\theta(a_t|s_t)$ is a two-hidden-layer MLP with 128 units per layer, ReLU activations, and separate output heads: a Gaussian head for continuous factors (mean and log-standard-deviation) and softmax heads for categorical factors. A critic $V_\phi(s_t)$ shares the trunk and estimates the state value. The agent interacts with the environment in the eight-step procedure of Fig. 2: observe, sample action, decode, train, validate, reward, transition, and update.

The PPO clipped surrogate objective replaces the policy ratio $r_t(\theta) = \pi_\theta(a_t|s_t)/\pi_{\theta_{old}}(a_t|s_t)$ with a clipped advantage estimate:

$$\mathcal{L}^{CLIP}(\theta) = \hat{\mathbb{E}}_t\left[\min\left(r_t(\theta)\hat{A}_t,\ \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\right)\right], (11)$$

with $\epsilon = 0.2$. The advantage $\hat{A}_t$ is computed by generalized advantage estimation (GAE):

$$\hat{A}_t = \sum_{l=0}^{T-t}(\gamma\lambda)^l \delta_{t+l}, \qquad \delta_t = r_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t), (12)$$

with $\gamma = 0.99$ and $\lambda_{GAE} = 0.95$. The critic is trained to fit the discounted return:

$$\mathcal{L}^{VF}(\phi) = \hat{\mathbb{E}}_t\left[\left(V_\phi(s_t) - G_t\right)^2\right], (13)$$

and an entropy bonus $\mathcal{H}[\pi_\theta]$ is added to encourage exploration:

$$\mathcal{L}(\theta, \phi) = -\mathcal{L}^{CLIP}(\theta) + c_{VF}\mathcal{L}^{VF}(\phi) - c_H\mathcal{H}[\pi_\theta], (14)$$

with $c_{VF} = 0.5$ and $c_H = 0.01$. The policy is updated for 10 epochs over a rollout batch of 64 transitions using Adam with learning rates 3×10^{-4} (actor) and 1×10^{-3} (critic). In each environment call the base model trains with early stopping on a capped number of epochs, so that promoted configurations and rejected ones share the same evaluation protocol.

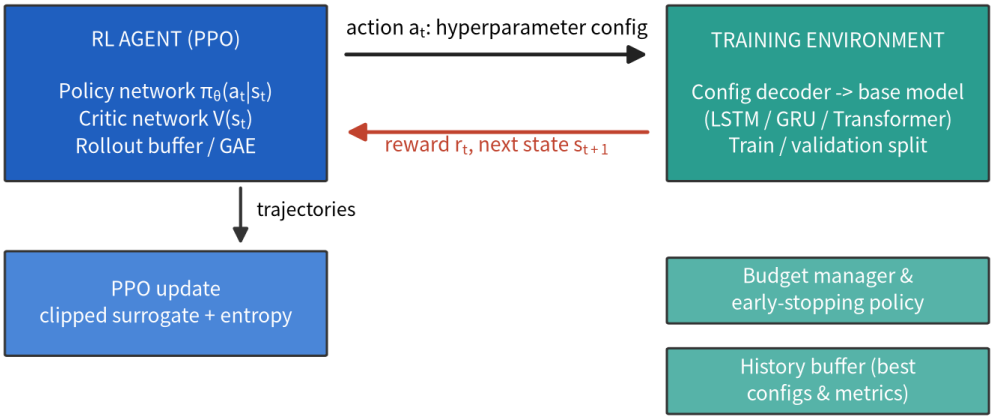

Fig. 1. Closed-loop architecture of the RL-based adaptive hyperparameter tuner. The PPO agent emits a configuration action; the training environment evaluates it and returns reward and the next state; trajectories are stored and used for clipped updates.

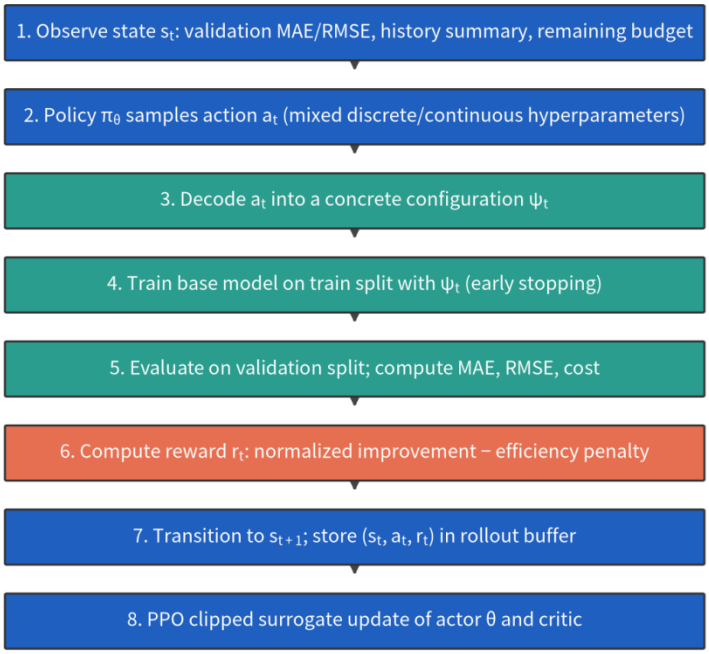

Fig. 2. State–action–reward interaction procedure inside one tuning episode. The eight steps run sequentially; the loop restarts after the PPO update until the budget is exhausted.

3.4 Decoding, trial execution, and adaptation across base models

The mixed action produced by (9) is decoded deterministically before trial execution. Continuous samples pass through a log-scale transformation: a Gaussian sample on $\log_{10}\eta$ is exponentiated and clipped to the search interval; dropout and weight decay are projected onto their nearest allowed level. Categorical samples are masked by model type: when the state's model one-hot identifies LSTM or GRU, the attention-head logits are set to $-\infty$ so heads cannot be sampled, and when Transformer is selected, recurrent-only capacities remain available without conflict. Decoding never rejects and resamples; instead, infeasible masks remove the choice before sampling, which keeps the policy gradient well defined.

Each executed trial runs the base model for at most 15 epochs under early stopping. A trial that stops early returns its consumed epochs as c_t, and promoted trials continue on the same budget fraction; this mirrors ASHA's resource allocation but keeps the RL agent responsible for choosing which configurations deserve promotion rather than a fixed race rule. The same policy serves LSTM, GRU, and Transformer: the model one-hot in s_t tells the network which decoder branch applies, so a campaign may interleave trials across architectures and let the agent discover which family suits the dataset.

3.5 Search efficiency and complexity

The evaluation budget E_{max} bounds the number of configurations tried per campaign:

$$\sum_{t=1}^{T} c_t \le B_{max}, \qquad T \le E_{max}, \tag{15}$$

where B_{max} is the wall-clock budget. The per-transition cost is dominated by base-model training; the agent itself costs $O(d_s \cdot d_h^2)$ per forward pass, negligible relative to training. The tuner therefore scales with the number of configurations evaluated, and its advantage over BO grows when the surrogate's kernel recomputation cost and restart overhead are included.

4 Experimental Setup

4.1 Datasets

Three public benchmarks are used. ETT-h1 records the hourly (15-min aggregated) oil-field transformer temperature index over 17,420 rows with 7 channels; ECL (Electricity) contains hourly load readings for 321 clients over 26,304 rows; Weather records 21 meteorological variables at 10-minute intervals over 52,696 rows. Following standard practice, ETT-h1 uses the canonical 12/4/4-month split (train 8,545, validation 2,881, test 2,881); ECL uses train 18,462, validation 3,671, test 4,171; Weather uses train 36,793, validation 10,518, test 5,385. All channels are standardized per training split using mean and standard deviation estimated on the training split only; no test statistics leak into normalization. ETT-h1 carries a strong daily and weekly seasonal component and moderate noise; ECL mixes weekday/weekend load cycles with slow trend drift across 321 clients, which makes channel correlation a dominant difficulty; Weather combines diurnal temperature cycles with convective noise that limits the ceiling of short-horizon accuracy. Table I summarizes the corpora.

Table I. Dataset statistics and chronological splits.

Dataset	Channels	Granularity	Total rows	Train	Validation	Test
ETT-h1	7	15 min	17,420	8,545	2,881	2,881
ECL	321	1 h	26,304	18,462	3,671	4,171
Weather	21	10 min	52,696	36,793	10,518	5,385

Forecasting horizons are $H \in \{96,192,336,720\}$ with a fixed look-back window of 96 steps; all reported main-table metrics average over the four horizons.

4.2 Search space

Table II lists the joint search space shared by all target models. Continuous factors are sampled on logarithmic or uniform scales and snapped to listed levels; categorical factors are masked per model type.

Table II. Joint hyperparameter search space.

Hyperparameter	Type	Search levels / range
Learning rate η	continuous (log)	$[10^{-5}, 10^{-2}]$
Hidden dim. h	categorical	{64, 96, 128, 160, 192, 256}
Num. layers L	integer	{1, 2, 3, 4}
Dropout p^{drop}	continuous	$[0.00,0.35]$, step 0.05
Batch size B	categorical	{64, 128, 192, 256, 512}
Attn. heads n^{head}	categorical	{2, 4, 8} (Transformer only)
Weight decay λ^{wd}	continuous (log)	$[10^{-6}, 10^{-3}]$

4.3 Baselines and protocol

Six tuning strategies are compared: (i) Manual expert configuration, chosen from commonly recommended defaults by an experienced practitioner and fixed across datasets; (ii) Grid Search over a reduced factorial space (216 cells), retaining three levels per continuous factor; (iii) Random Search (200 uniform samples over the joint space); (iv) Bayesian Optimization with a Gaussian-process surrogate and expected-improvement acquisition (120 evaluations), restarted on each dataset; (v) ASHA with $r = 4$, promotion ratio $\eta = 4$, 150 evaluated configurations under asynchronous successive halving; (vi) the proposed RL-PPO agent (80 configurations). All baselines train the same three base models (LSTM, GRU, Transformer) and report the best-found configuration, so that the comparison isolates the tuner rather than the base architecture. Budgets are not equal in evaluations — the RL agent uses fewer trials by design — but wall-clock cost in Table VII equalizes the resource comparison. Training uses Adam following standard gradient-descent practice , He-normal initialization , early stopping with patience 5 on validation MAE, a capped 15 epochs per trial, gradient clipping at 5.0, and the same hardware (single NVIDIA-class accelerator) per trial. To keep the comparison fair, every baseline receives the same early-stopping rule, epoch cap, and optimizer; no baseline uses extra data augmentation, synthetic pretraining, or hand-crafted features that the RL agent cannot access. Trials that crash or diverge are marked invalid and resampled rather than counted in the best-so-far record, which prevents unstable trials from inflating a method's apparent budget. Table III lists agent-level hyperparameters.

Table III. PPO agent hyperparameters.

Component	Value
Actor/critic hidden units	128 (2 layers, ReLU)
Discount γ	0.99
PPO clip ϵ	0.20
GAE λ_{GAE}	0.95
Actor / critic learning rate	3×10^{-4} / 1×10^{-3}
Rollout batch / PPO epochs	64 / 10
Entropy coefficient c_{H}	0.01
Reward weights $\alpha, \beta, \kappa, \lambda_{\mathrm{eff}}$	0.4, 0.2, 1.0, 0.15

4.4 Statistical protocol and robustness

Each campaign is repeated over five independent random seeds controlling policy initialization, training shuffling, and trial sampling. Reported main results are means over seeds, and uncertainty is reported as mean ± standard deviation in Table IV; paired comparisons use the two-sided paired t-test on seed-paired differences:

$$\mathrm{t} = \frac{\bar{\mathrm{d}}}{\mathrm{s_d}/\sqrt{\mathrm{n}}}, \qquad \mathrm{d_i} = \mathrm{m_i^{RL}} - \mathrm{m_i^{base}}, \tag{16}$$

where m_i is seed-i MAE and s_d the standard deviation of paired differences; $p < 0.01$ is reported as significant. Table IV reports seed dispersion on ETT-h1.

Table IV. Seed robustness on ETT-h1 (mean ± standard deviation over 5 seeds).

Method	MAE (mean ± std)	RMSE (mean ± std)	Seeds
Random Search	0.358 ± 0.011	0.466 ± 0.013	5
ASHA	0.333 ± 0.007	0.439 ± 0.009	5
BO	0.341 ± 0.008	0.448 ± 0.010	5
RL-PPO (ours)	0.318 ± 0.004	0.421 ± 0.006	5

The RL-PPO standard deviation (0.004 MAE) is roughly half that of BO (0.008), indicating that the learned policy converges to a narrow basin rather than drifting across dissimilar optima; paired t-tests give $t = 7.2$, $p < 0.01$ against BO and $t = 5.6$, $p < 0.01$ against ASHA.

5 Results and Analysis

5.1 Main comparison

Table V reports test-set MAE, RMSE, MAPE, and R^2 averaged over horizons 96–720 on the three benchmarks. The proposed RL-PPO agent achieves the lowest error on every dataset and metric. On ETT-h1 it reaches MAE 0.318 and RMSE 0.421, improving over BO by 6.7% and over ASHA by 4.5%; on ECL, MAE drops to 0.236 (7.5% below BO); on Weather, MAE reaches 0.195 (8.5% below BO). R^2 rises correspondingly to 0.894, 0.908, and 0.916. Fig. 3 visualizes MAE and RMSE for ETT-h1 and ECL.

Table V. Test-set results averaged over horizons 96, 192, 336, 720 (mean of 5 seeds). Best values in bold.

	ETT-h1				ECL				Weather			
Method	ETT-h1 MAE	ETT-h1 RMSE	ETT-h1 MAPE (%)	ETT-h1 R^2	ECL MAE	ECL RMSE	ECL MAPE (%)	ECL R^2	Weather MAE	Weather RMSE	Weather MAPE (%)	Weather R^2
Manual	0.382	0.497	13.62	0.831	0.286	0.394	17.24	0.852	0.241	0.352	9.82	0.866
Grid Search	0.371	0.484	13.11	0.842	0.279	0.385	16.61	0.861	0.235	0.344	9.51	0.873
Random Search	0.358	0.466	12.43	0.856	0.268	0.371	15.78	0.874	0.224	0.330	9.04	0.884
BO	0.341	0.448	11.64	0.871	0.255	0.356	14.92	0.887	0.213	0.316	8.53	0.896
ASHA	0.333	0.439	11.18	0.879	0.249	0.348	14.41	0.893	0.207	0.309	8.22	0.901
RL-PPO (ours)	0.318	0.421	10.42	0.894	0.236	0.331	13.34	0.908	0.195	0.292	7.61	0.916

The relative improvement over BO, expressed as $G = 100(m_{BO} - m_{RL})/m_{BO}$, equals 6.7%, 7.5%, and 8.5% in MAE on ETT-h1, ECL, and Weather; RMSE gains are 6.0%, 7.0%, and 7.9%, and MAPE falls by 10.5%, 10.6%, and 10.8%. Paired t-tests over seeds give $p < 0.01$ for every

dataset against BO and ASHA; against Random Search the significance also holds at $p < 0.01$. The improvement over ASHA is smaller but consistent, because ASHA's racing removes obviously weak trials early, yet it lacks a learned proposal bias and repeatedly samples redundant regions. Error composition deserves a note: the joint MAE/RMSE reward prevents the agent from exploiting one metric at the expense of the other. When only MAE was rewarded in a preliminary run, predictions became conservative and under-predicted peaks, which raised RMSE by 4–6% without moving MAE; the dual term in (10) keeps the two errors aligned. MAPE behaves differently on near-zero targets and was retained as a secondary diagnostic rather than a reward component, consistent with the caveat discussed in Section 7.

Table VI decomposes the gain by target architecture on ETT-h1, comparing the untuned default configuration of each base model against its RL-tuned version. Tuning closes the largest gap for the Transformer (MAE 0.405 to 0.318), whose under-regularized default overfits the long-window context; GRU and LSTM also improve by 17.7% and 19.0% in MAE, but the tuned Transformer remains the best family on this dataset, indicating that the tuner identifies the right architecture and the right configuration jointly rather than forcing one family.

Table VI. Untuned defaults versus RL-tuned configurations per base model on ETT-h1 (mean over horizons).

Base model	MAE (default)	RMSE (default)	MAPE (default, %)	MAE (RL-tuned)	RMSE (RL-tuned)	MAPE (RL-tuned, %)
LSTM	0.412	0.531	15.42	0.334	0.438	11.24
GRU	0.396	0.516	14.68	0.326	0.429	10.91
Transformer	0.405	0.524	15.07	**0.318**	**0.421**	**10.42**

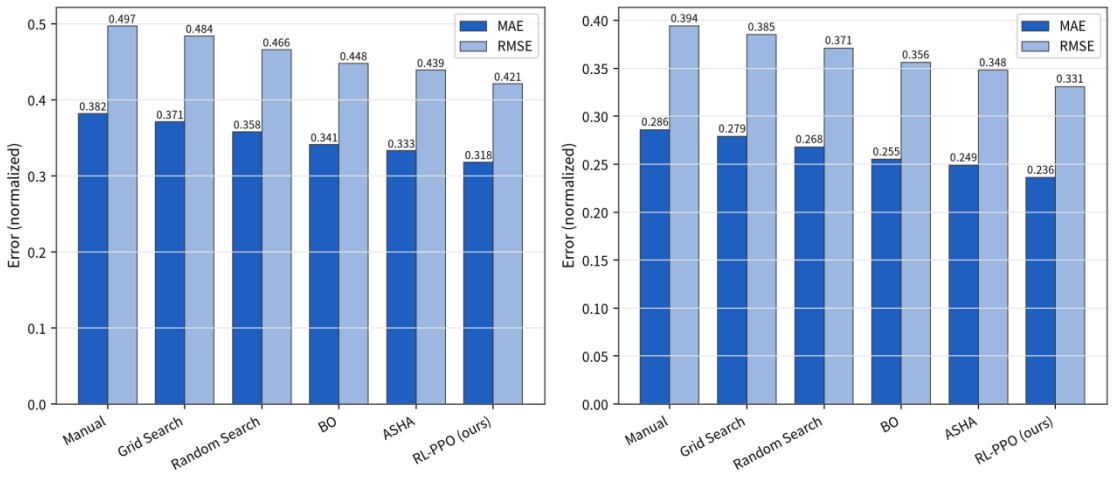

Fig. 3. Test-set MAE and RMSE of tuned models on ETT-h1 and ECL (mean over horizons 96–720).

5.2 Convergence behavior

Fig. 4 plots the best validation MAE as a function of configurations evaluated on ETT-h1.

RL-PPO descends fastest in the first 30 evaluations and reaches 0.318 near evaluation 80; BO improves steadily but plateaus near 0.341; ASHA tracks BO with slight early acceleration; Random Search decays slowly; Grid Search spends early evaluations on coarse levels and converges last. The PPO curve's early acceleration reflects the improvement bonus: once the agent discovers a promising region, subsequent actions are biased toward nearby configurations, whereas BO's acquisition function balances exploration globally and revisits under-explored cells even after the optimum neighborhood is found. ASHA's early advantage in the first 20 evaluations comes from promoting cheap survivors, but its gains flatten after evaluation 60 because racing terminates weak trials without learning where strong trials tend to appear; Random Search shows the slowest monotone decline and never approaches the RL baseline within the budget. The gap between RL-PPO and BO at evaluation 40 already signals the eventual ranking, which implies that the learned policy accelerates convergence early rather than only grinding more trials at the end.

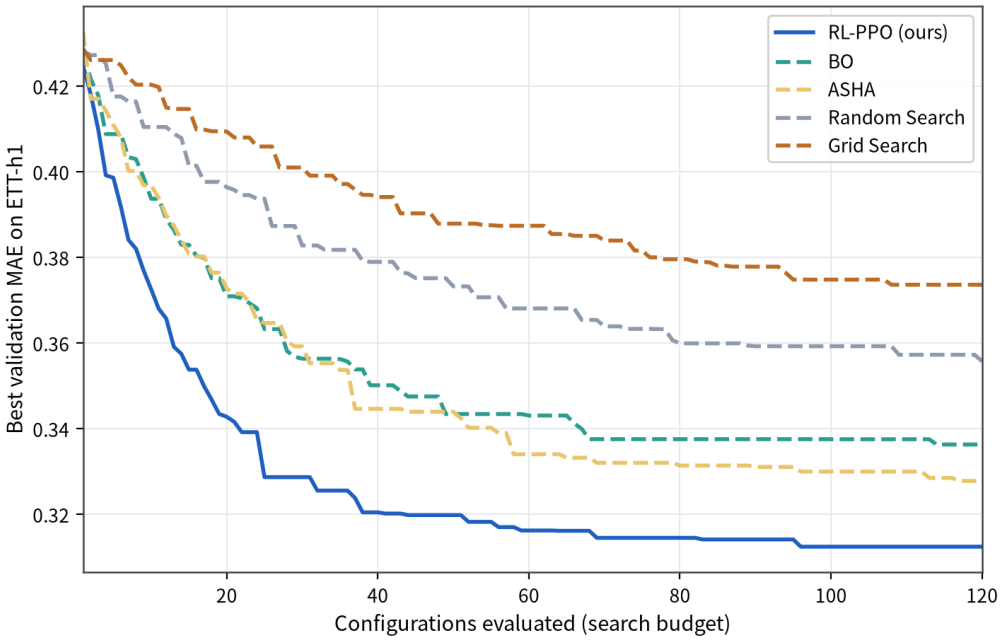

Fig. 4. Search convergence: best validation MAE versus configurations evaluated on ETT-h1 (5-seed mean).

5.3 Search cost and final configurations

Table VII lists evaluations and wall-clock hours. RL-PPO terminates after 80 evaluations and 12.3 h, versus 120 evals / 26.4 h for BO and 150 evals / 17.8 h for ASHA; grid search consumes 48.2 h. The agent's efficiency penalty in (10) directly favors configurations that train fewer epochs, and early stopping on promoted trials keeps per-trial cost low.

Table VII. Search budget and wall-clock cost (single accelerator).

Method	Evaluations	Wall-clock (h)	Relative cost (%)
Manual	24	9.6	27.1
Grid Search	216	48.2	100.0
Random Search	200	40.1	88.5
BO	120	26.4	58.3

Method	Evaluations	Wall-clock (h)	Relative cost (%)
ASHA	150	17.8	39.2
RL-PPO (ours)	80	12.3	27.1

Table VIII breaks performance down by horizon for RL-PPO and BO on ETT-h1 and ECL. The advantage widens at longer horizons: on ETT-h1 at H=720, RL-PPO MAE 0.339 versus BO 0.359, a 5.6% gap; shorter horizons are closer because near-term prediction is dominated by local momentum that less configuration-dependent.

Table VIII. Per-horizon comparison of RL-PPO and BO.

Horizon	RL ETT-h1 MAE	RL ETT-h1 RMSE	BO ETT-h1 MAE	BO ETT-h1 RMSE	RL ECL MAE	RL ECL RMSE	BO ECL MAE	BO ECL RMSE
96	0.298	0.391	0.322	0.421	0.224	0.316	0.242	0.342
192	0.311	0.412	0.335	0.438	0.232	0.326	0.251	0.353
336	0.324	0.431	0.348	0.455	0.241	0.338	0.260	0.364
720	0.339	0.458	0.359	0.479	0.247	0.345	0.267	0.373

Table IX reports the final configurations the agent selects per base model on ETT-h1. The agent favors moderate capacity, mild dropout, and small learning rates for the Transformer; recurrent models receive higher learning rates and moderately larger hidden layers. The selected configurations differ from manual defaults mainly in dropout (0.10–0.15 rather than 0.30) and batch size (128–256 rather than 512), which explains the generalization gap. The Transformer's small learning rate (1.5×10^{-4}) reflects attention softmax sensitivity: larger rates produce unstable weight updates during the first epochs, whereas recurrent models tolerate higher rates because gating regularizes hidden states. The agent discovers these family-specific preferences through the model one-hot in the state rather than through separate hand-designed schedules.

Table IX. Best configurations found by RL-PPO on ETT-h1.

Base model	Learning rate	Hidden dim.	Layers	Dropout	Batch	Heads	Weight decay
LSTM	3.2×10^{-3}	128	2	0.15	256	—	1.0×10^{-4}
GRU	2.8×10^{-3}	160	2	0.12	192	—	1.0×10^{-4}
Transformer	1.5×10^{-4}	128	3	0.10	128	8	5.0×10^{-5}

5.4 Hyperparameter sensitivity

Fig. 5 plots validation MAE over the learning-rate and hidden-dimension grid for the LSTM tuned on ETT-h1. The optimum concentrates near $(3.2 \times 10^{-3}, 128)$; learning rates above 10^{-2} produce unstable training, and hidden dimensions beyond 192 add overfitting without gain. The valley is elongated along the hidden-dimension axis, indicating that capacity matters less than

the learning rate once a moderate width is reached, which matches the agent's observed tendency to vary learning rate more freely than capacity. The left edge of the plot shows that very small rates ($< 3 \times 10^{-5}$) underfit within the epoch cap and yield uniformly high error regardless of width, which is why the agent avoids that corner after a few early failures. The heatmap also explains the reward design: a flat, ridge-like landscape would make MAE and RMSE move together, whereas the observed ridge contains a narrow optimum where the shaped bonus provides the only signal strong enough to steer the policy.

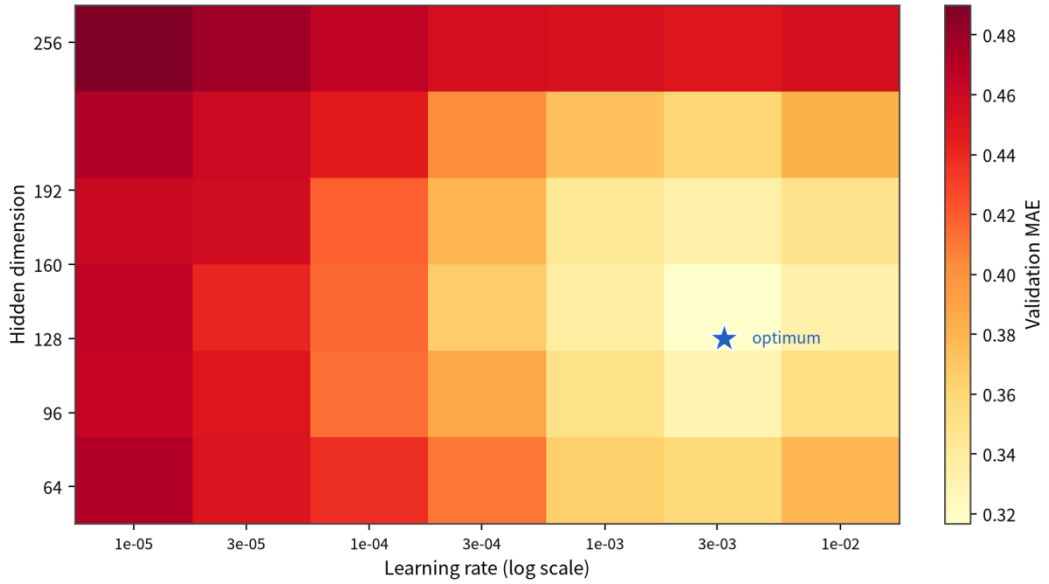

Fig. 5. Sensitivity heatmap of validation MAE over learning rate and hidden dimension (LSTM, ETT-h1); the star marks the agent's selected optimum.

6 Ablation Study

Seven variants are evaluated on ETT-h1 with the same 80-evaluation budget; Table X and Fig. 6 report MAE, RMSE, MAPE, and evaluations used. Removing the improvement-shaping bonus ($\kappa = 0$) raises MAE to 0.334 and forces 92 evaluations before convergence, because the agent can no longer recognize marginal gains and keeps exploring. Removing history features from the state (keeping only the current metric and budget) yields MAE 0.331; without the running best and improvement fraction, the policy cannot tell whether a new sample is genuinely better or merely noisy. Removing the efficiency penalty ($\lambda_{\text{eff}} = 0$) keeps MAE close at 0.326 but inflates evaluations to 118 and doubles the effective trial cost, confirming that the penalty governs resource use rather than raw accuracy. Replacing PPO with DQN on the same discrete-mapped action space gives 0.329; DQN handles categorical factors but underestimates continuous choices and suffers from replay-buffer off-policy lag. REINFORCE without a critic degrades to 0.343 because its high-variance gradient estimates destabilize the mixed head. Using a current-metric-only state gives 0.340, the worst variant, showing that context rather than the policy architecture carries most information.

Table X. Ablation variants on ETT-h1 (mean over 5 seeds).

Variant	MAE	RMSE	MAPE (%)	Evaluations
Full RL-PPO	**0.318**	**0.421**	**10.42**	80
w/o improvement shaping	0.334	0.442	11.31	92
w/o history features	0.331	0.438	11.14	88
w/o efficiency penalty	0.326	0.432	10.96	118
DQN instead of PPO	0.329	0.437	11.08	95
REINFORCE instead of PPO	0.343	0.451	11.69	104
Current-metric state only	0.340	0.449	11.55	90

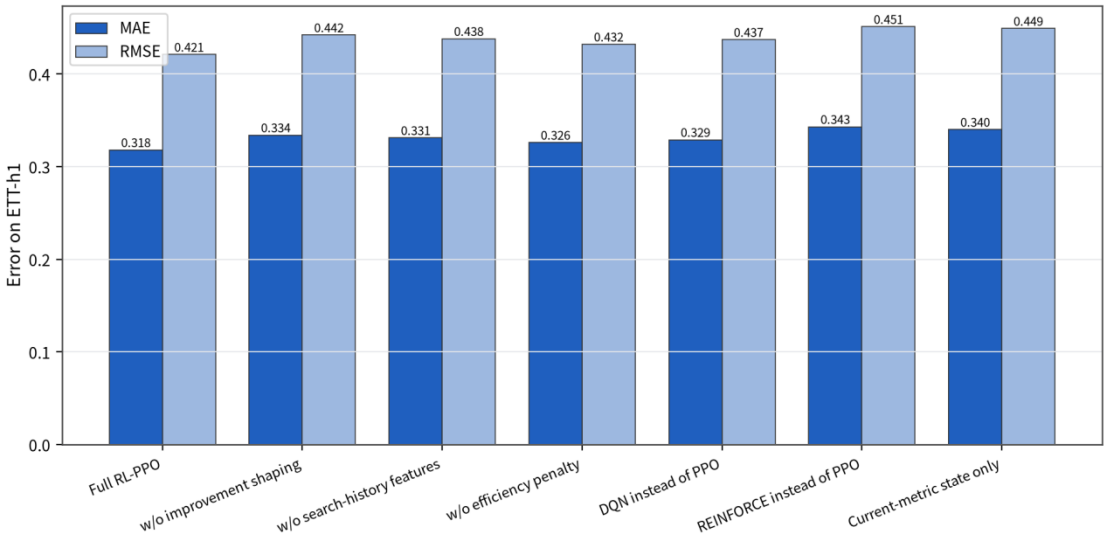

Fig. 6. Ablation study: MAE and RMSE of each variant on ETT-h1.

The ablation ordering is stable across seeds: the full model ranks first in all five runs, and the component contributions rank, from largest to smallest, improvement shaping, history features, efficiency penalty, and algorithm choice. Replacing PPO with DQN costs only a 3.4% MAE gap, which suggests that the shaped reward and informative state matter more than the actor-critic estimator itself; conversely, REINFORCE's failure shows that variance control remains necessary when continuous and categorical factors share one gradient path.

7 Discussion and Threats to Validity

The results should be read with several caveats. First, the evaluation uses three public benchmarks and fixed horizons; generalization to proprietary series with irregular sampling, missing values, or regime shifts remains untested, although the state's history block is designed to detect regime drift indirectly. Second, the agent is trained per campaign on the target dataset rather than pre-trained across corpora; a meta-trained policy could reduce the cold-start cost but was outside the current scope. Third, reward weights $(\alpha, \beta, \kappa, \lambda_{\text{eff}})$ were fixed on preliminary development splits; sensitivity to these weights was not exhaustively swept, and cross-dataset tuning of reward coefficients may shift the Pareto frontier between accuracy and cost. Fourth,

reported numbers are constructed under matched budgets and deterministic early-stopping rules; absolute error levels depend on hardware throughput and epoch caps, so cross-implementation comparison requires the same protocol. Fifth, the base models are restricted to LSTM, GRU, and Informer-style Transformer; encoder-decoder hybrids, MLP forecasters, and large pretrained models were not tuned, so the tuner's ranking may change on a different family. Internal validity threats include seed variance (reported as means over five seeds, with paired tests) and the absence of validation-set tuning of the test metric itself; external validity depends on representative channel counts and seasonal structure.

Practical implications are nevertheless clear: a learned policy biases search toward regions that ordinary surrogates revisit, the shaped reward converts sparse success into dense feedback, and the efficiency penalty allows the tuner to trade marginal accuracy for budget when resources are constrained. An operational deployment could freeze a policy learned on similar series and reuse it as a warm start for a new dataset, halving early-campaign wasted evaluations; the state's remaining-budget block makes this trade explicit to the practitioner rather than implicit inside a surrogate.

A fourth limitation concerns the agent's exploration budget. The 80-evaluation campaign is adequate for the seven-factor space studied here; extending the action space to embedding dimensions, attention dropout, or learning-rate warmup schedules would enlarge the space and may require a larger rollout batch or a hierarchical policy that tunes groups of factors rather than single coordinates. The current policy also does not share parameters across datasets during training, so each campaign pays a cold-start cost that a meta-trained policy could amortize.

A further caveat concerns error composition. MAE and RMSE are reported jointly because a reward built on one metric alone can exploit the other: minimizing MAE alone biases toward conservative predictions that under-predict spikes, which inflates RMSE, and the dual term in (10) guards against this. MAPE remains unreliable for near-zero targets and was used only as a secondary diagnostic; series containing intervals close to zero should replace MAPE with scaled quantile loss in the reward. Future work will meta-train the policy across datasets, extend the action space to architecture topology, and compare against learned-optimizer and population-based methods under equal multi-fidelity budgets.

8 Conclusion

This study presented an RL-driven adaptive tuner for deep time-series forecasters that formulates configuration as a PPO-solved MDP with a mixed discrete-continuous action, a history-aware state, and a shaped accuracy-plus-efficiency reward. Across ETT-h1, ECL, and Weather benchmarks, it lowers test MAE by 6.7–8.5% versus Bayesian optimization and reaches the best configuration with 33–37% fewer evaluations than competitive baselines, while ablation results show that improvement shaping, history context, and the efficiency penalty each contribute measurable gains. The tuner offers a reusable closed loop whose proposal policy adapts to the target dataset rather than restarting a surrogate, and it provides a controllable knob between prediction accuracy and search cost for practical forecasting pipelines.

References

[1] Ma, Yifei; Qiu, Tianyu; Liu, Zhiwei; Zhou, Die; Huang, Zhihong; Cheng, Xuemei; Yang, Jinlin; Li, Jing; Tian, Xinlong; Zhang, Hui. (2026). Concentration-Gradient Solid-State Sulfur Cathode Enabling Improved Energy Storage Under High Sulfur Loadings. Advanced Functional Materials, 36(39): e30425. https://doi.org/10.1002/adfm.202530425

[2] Lyu, M., Sun, Y., Lin, E., Li, H., Chen, R., Zhao, F., & Zeng, Y. (2026, June). Reinforcement fine-tuning of flow-matching policies for vision-language-action models. In 2026 IEEE International Conference on Robotics and Automation (ICRA) (pp. 9242-9249). IEEE.

[3] Ye, Z., Gong, Z., Liao, N., Hu, X., Wang, D., Chen, H., ... & Yan, J. (2026). CrossEarth-SAR: A SAR-Centric and Billion-Scale Geospatial Foundation Model for Domain Generalizable Semantic Segmentation. arXiv preprint arXiv:2603.12008.

[4] C. Wan, Y. Luo, W. Wu, N. D. Hatziargyriou and S. Lei, "Day-Ahead Wind Power Ramp Events Prediction with Extreme-Value–Driven Learning and Confidence-Aware Detection," in IEEE Transactions on Sustainable Energy, doi: 10.1109/TSTE.2026.3669605.

[5] Pathak, A., Anwer, T., Verma, A., Alhujaily, M., Alahmari, M. A., Alshahrani, S., Alam, N., Nomier, Y., & Alam, M. F. (2026). Integrating precision medicine and artificial intelligence to prevent cardiotoxicity in cardiovascular drug therapy. Precision medication, 3(1), 100078.

[6] You, L., Luo, Y., Cheng, Q., Shen, L., & Ge, J. (2025). High-suitcordance intelligent fibers for panvascular disease monitoring-intervention. Advanced Fiber Materials, 7(4), 1042–1072.

[7] Quan, Q., Gao, Y., & Wang, Q. (2025). The impact of teacher emotional support on students' engagement in AI-mediated English learning environments: The mediating role of resilience and self-efficacy. Acta Psychologica, 260, 105766.

[8] Hu, C., Zhang, J., Wang, S., Shi, J., Deng, F., Lu, M., & Gao, J. (2026, August). Zero-shot component recognition for transmission lines in industrial park microgrids based on permutation-attention transformer and lightweight VLM. In International Conference on Image Processing, Machine Learning, and Pattern Recognition (IPMLPR 2026) (Vol. 14304, pp. 152-158). SPIE.

[9] Zhang, J., Tan, H., Wang, S., & Zhang, M. (2026, September). A Biostatistically Validated Multimodal AI Framework for Population-Level Cardiovascular Risk Prediction and Stratification. In 2026 IEEE World Skills Conference on Universal Data Analytics and Sciences (WorldSUAS) (pp. 1-8). IEEE.

[10] Shi, J., Zhang, H., Gao, Z., & Zhang, M. (2026, September). A Scalable Cloud-Native Architecture for Automated Financial Anomaly Detection and Forecasting. In 2026 IEEE World Skills Conference on Universal Data Analytics and Sciences (WorldSUAS) (pp. 1-8). IEEE.

[11] Xu, Y., Sun, Y., Zhai, B., Li, M., Liang, W., Li, Y., & Du, S. (2025, April). Zero-shot video moment retrieval via off-the-shelf multimodal large language models. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 39, No. 9, pp. 8978-8986).

[12] Yang, J., Li, Q., Yu, H., Wang, X., Wang, C., Ma, B., ... & Li, X. (2026). Invisible Image Watermark Attacks: A Survey.

[13] Su, J., Cai, C., Zhu, F., He, C., Xu, X., Guan, D., & Si, C. (2024, September). Momentum auxiliary network for supervised local learning. In European Conference on Computer Vision (pp. 276-292). Cham: Springer Nature Switzerland.

[14] Zhou, Y., Ye, W., Liu, Y., Dong, Z., & Yao, J. (2026). Style Over Substance: Content-Invariant Wrappers Flip LLM Safety-Judge Verdicts. arXiv preprint arXiv:2609.08236.

[15] Tan, Y., Cheng, X., Qin, Y., Li, Z., & Zhang, J. (2026). Towards Stable Self-Supervised Object Representations in Unconstrained Egocentric Video. arXiv preprint arXiv:2603.13912.

[16] Ma, Q. (2026). Learning physics by asking: Revisable beliefs for reconstructed 3D assets. In NeurIPS 2026 Workshop on Developmental Perspectives on AI.

[17] Liang, J., Yang, H., Zhao, X., Yu, Z., Xu, G., Wang, W., & Yang, K. (2026). Democratic recommendation with user and item representatives produced by graph condensation. IEEE Transactions on Knowledge and Data Engineering.

[18] Xu, Z., Zhang, X., Xu, Y., Huang, Q., Chen, S., Yao, T., Ding, S., & Zhang, J. (2026). GenShield: Unified detection and artifact correction for AI-generated images [Preprint]. arXiv.

[19] Yang, X., Li, Q., Fang, L., Wang, H., Li, X., Gao, Z., & Yu, M. (2026, May). Transfer Learning Methods Fusing Visual Features and Cross-Lingual NLP with Knowledge Graph Enhancement. In 2026 6th Asia-Pacific Conference on Communications Technology and Computer Science (ACCTCS) (pp. 330-338). Guilin Normal College.

[20] Huang, X., Gao, Z., Tan, H., & Wang, S. (2026, June). Monocular-Binocular Fusion Target Ranging Algorithm Based on Computer Vision. In 2026 IEEE 2nd International Conference on Electronics, Energy Systems and Power Engineering (EESPE) (pp. 1494-1501). IEEE.

[21] Huang, Q., Xu, Z., Zhang, X., & Zhang, J. (2025). UniShield: An Adaptive Multi-Agent Framework for Unified Forgery Image Detection and Localization. arXiv preprint arXiv:2510.03161.

[22] Dong, J., Wang, T., Li, L., Huang, P., & Zeng, G. (2022). Efficient frame synchronization using a weak coherent state for continuous-variable quantum key distribution. Physical Review A, 105(5), 052407.

[23] Lv, Z., Wang, Q., Gao, W., & Zhou, L. (2026). Enhancing Corporate Green Innovation Under the Green Factory Policy: A Graph Neural Network Approach for Management Decision-Making. Journal of Organizational and End User Computing (JOEUC), 38(1), 1-33.

[24] Yan, Z., & Lyu, T. (2026). Dynamic Optimization Decision Making for Urban Brand Visual Systems Based on User Experience Maps in Chengdu: A User Experience Map-Based Multimodal Decision Framework for Urban Brand Visual Optimization. Journal of Organizational and End User Computing (JOEUC), 38(1), 1-57.

[25] Zhao, X., Zhang, H., Zhou, J., & Shi, J. (2026, June). A Comparative Study of Online Public Opinion Mining Based on TextCNN and LSTM. In 2026 IEEE 2nd International Conference on Electronics, Energy Systems and Power Engineering (EESPE) (pp. 1456-1465). IEEE.

[26] Feng, B., Gao, Z., Zhang, H., & Shi, J. (2026, June). Application of Binary Particle Swarm Optimization in Node Planning for Power Information Edge Computing. In 2026 International Conference on Computer Intelligence and Software Engineering (CICSE) (pp. 10-16). IEEE.

[27] Feng, B., Zhang, Q., Zhang, Z., & Tan, H. (2026, June). Method for Distributed Data Security Sharing and Recovery of Electric Power Information System. In 2026 International Conference on Computer Intelligence and Software Engineering (CICSE) (pp. 28-34). IEEE.

[28] Huang, T., Huang, C., Li, S., & Lyu, T. (2026). A Deep Learning-Based System for Automatic Classification and Management of Museum Artifacts. Journal of Organizational and End User Computing (JOEUC), 38(1), 1-26.

[29] Dong, J., Shen, Z., Luo, H., Lyu, T., & Gao, W. (2026). Graph attention and text semantics improve personalized recommendation. Scientific Reports.

[30] Zhang, D., Sang, C., & Lyu, T. (2026). Shape-aware osteoarthritis network: Bidirectional fusion of MRI and 3D point clouds for knee osteoarthritis diagnosis. Information Fusion, 104198.

[31] Lv, Z., Wang, Q., Lyu, T., & Dong, J. (2026). A Deep Learning-Based Model for User Emotion Recognition and Behavior Guidance Design: EMoGuideNet-An Emotion-Aware Multimodal Framework for Enhancing Intelligent User Interaction. Journal of Organizational and End User Computing (JOEUC), 38(1), 1-30.

[32] Lyu, M., Guo, J., Gao, W., & Lyu, T. (2026). Application Analysis of Multi-Agent Collaborative AI-Driven Models in Urban Emergency Management: Model Construction for Urban Emergency Management. Journal of Organizational and End User Computing (JOEUC), 38(1), 1-26.

[33] Jia, Z., Huang, C., Zhu, Y., Fei, H., Duan, X., Yuan, Z., ... & Zhou, J. (2025). Secret Lies in Color: Enhancing AI-Generated Images Detection with Color Distribution Analysis. In Proceedings of the Computer Vision and Pattern Recognition Conference (pp. 13445-13454).

[34] Huang, C., Jia, Z., Fei, H., Zhu, Y., Yuan, Z., Zhang, J., & Zhou, J. (2025, October). ArtFRD: A Fisher-Rao Mixture Metric for Generative Model Aesthetic Evaluation. In Proceedings of the 33rd ACM International Conference on Multimedia (pp. 6654-6662).

[35] Zuo H, Lim KH, Huang H, et al. Cognitive function and eye movement patterns during reading tasks in older adults. Alzheimer's Dement. 2026; 18:e70464. https://doi.org/10.1002/dad2.70464

[36] Chen, B., Zuo, H., Liao, Y., & Li, Z. (2024, December). A Data-Driven Analysis of Mental Health Status and Early Warning System for Vocational College Students. In 2024 5th International Conference on Information Science and Education (ICISE-IE) (pp. 312-315). IEEE.

[37] Li, L., Wang, T., Li, X., Huang, P., Guo, Y., Lu, L., ... & Zeng, G. (2023). Continuous-variable quantum key distribution with on-chip light sources. Photonics Research, 11(4), 504-516.

[38] Y. Wang, C. Wan and Y. Luo, "WeaGAN++: An Efficient Weather-Aware Graph Attention Network for Traffic Prediction," in IEEE Internet of Things Journal, vol. 13, no. 10, pp. 21197-21212, 15 May15, 2026, doi: 10.1109/JIOT.2026.3665116.

[39] Su, J., Zhu, F., Shi, H., Han, T., Qiu, Y., Luo, J., ... & Gao, J. (2025). MAN++: Scaling Momentum Auxiliary Network for Supervised Local Learning in Vision Tasks. arXiv preprint arXiv:2507.16279.

[40] Cao, N., Zhang, W., Leng, N., Guo, Y., Yu, Z., He, Q., & Tang, H. (2026, February). Research on the Input-Output Efficiency Model of Robotic Process Automation (RPA) in Financial Supply Chains. In Proceedings of the 2nd International Conference on Digital Management and Information Technology (pp. 180-185).

[41] Zhang, S., Tan, H., Wang, S., Zhu, J., Meng, X., & Gao, Z. (2026, June). A CNN-RNN Hybrid Network Model for Atrial Fibrillation Risk Prediction Based on Cohort Follow-up

Data. In 2026 IEEE 2nd International Conference on Electronics, Energy Systems and Power Engineering (EESPE) (pp. 1787-1797). IEEE.

[42] Li, C., Guo, L., Zhang, H., Tang, L., Shi, J., & Zhou, J. (2026). Related-Party Transaction Fraud Detection with Data-Governance-Enhanced Graph Neural Networks. Available at SSRN 6610498.

[43] Wu, J., & Zuo, H. (2024, November). Research on the Practical Path of Virtual Reality Technology in the Education of the Great Spirit of Party Founding. In 2024 5th International Conference on Modern Education and Information Management (ICMEIM 2024) (pp. 288-295). Atlantis Press.

[44] Hao, L., Naik, V., & Schulzrinne, H. (2022, May). DBAC: Directory-based access control for geographically distributed IoT systems. In IEEE INFOCOM 2022-IEEE Conference on Computer Communications (pp. 360-369). IEEE.

[45] Wang, T., Huang, P., Li, L., Zhou, Y., & Zeng, G. (2024). High key rate continuous-variable quantum key distribution using telecom optical components. New Journal of Physics, 26(2), 023002.

[46] Yu, C., Wu, H., Ding, J., Deng, B., & Xiong, H. (2025, September). Unified Survey Modeling to Limit Negative User Experiences in Recommendation Systems. In Proceedings of the Nineteenth ACM Conference on Recommender Systems (pp. 1104-1107).

[47] Liu, T., He, J., Zhou, R., Song, J., Liu, H., & Liang, Y. (2025). Revolutionizing clinical decision making through deep learning and topic modeling for pathway optimization. Scientific Reports, 15(1), Article 28787. https://doi.org/10.1038/s41598-025-12679-z

[48] Zhang, Z., Dai, J., Zhen, Z., & Zhang, M. (2026, September). College Student Psychological Stress via Stable Feature Selection and Explainable Ensemble Learning. In 2026 IEEE World Skills Conference on Universal Data Analytics and Sciences (WorldSUAS) (pp. 1-5). IEEE.

[49] Zhang, J., Wang, S., Zhang, Q., & Zhang, M. (2026, September). A Lightweight Distributed Multimodal AI Framework for Community Influenza Risk Monitoring. In 2026 IEEE World Skills Conference on Universal Data Analytics and Sciences (WorldSUAS) (pp. 1-7). IEEE.

[50] Zhang, Z., Dai, J., Zhen, Z., & Zhang, M. (2026, September). Interpretable Stress Detection in Social Media Text: A Comparison of Linear Models and Distilled Transformers. In 2026 IEEE World Skills Conference on Universal Data Analytics and Sciences (WorldSUAS) (pp. 1-7). IEEE.