FROM LOCAL ACCURACY TO GLOBAL FORECASTING: EXPERT SELECTION FOR MULTISTEP MULTIVARIABLE TOKAMAK FUSION DYNAMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Amid growing concerns over the global energy crisis and climate change, fusion energy has emerged as a promising route toward sustainable power, while artificial intelligence is increasingly being used to model and control tokamak plasmas. Yet its dynamics are high-dimensional, strongly coupled, and only partially observed, while practical forecasting requires multistep recursive prediction across heterogeneous physical variables. Existing deep learning predictors are difficult to achieve consistent and accurate predictions across all physical variables and time steps using a single model. Moreover, when using predictions as recursive inputs for future models, local accuracy may not necessarily determine the performance of multi-step predictions. Based on the above discovery, we propose that different models should be adaptively used for different time steps and physical variables, and systematically compared the four Mixture-of-Experts (MoE) frameworks in this work, including the behavior-cloning baseline (Baseline-MoE), DAgger-style MoE (DaD-MoE), reinforcement-learning-based MoE (PPO-MoE), and full-information rollout-aware MoE (FIRE) over a frozen bank of heterogeneous physical predictors. We first evaluate the proposed MoE framework through simulation and further extend them to real-world device measurements. The results show that PPO-MoE achieves the lowest complete-rollout error on simulated MAST-U vertical-displacement events, reducing it by 6.38% relative to Baseline-MoE. On experimental measurement from the real-world MAST device, FIRE reduces complete-rollout error by 12.29% relative to the best fixed-expert reference and by 7.81% relative to Baseline-MoE. These results demonstrate that adaptive expert routing can improve recursive multivariable plasma forecasting and offer a new perspective on forecasting in fusion-energy systems. Code is available at link.

1 INTRODUCTION

Fusion energy is being pursued as a potential source of abundant, reliable, and low-carbon electricity, while the tokamak remains one of the most mature and extensively studied fusion-energy plant concepts (Donné, 2019; Ongena et al., 2016; Meyer et al., 2013). However, a tokamak fusion-energy system is a strongly coupled dynamical system in which magnetic-field geometry, plasma equilibrium and transport, heterogeneous diagnostics, and actuator dynamics interact across multiple physical and temporal scales. Its evolving state is high-dimensional and only sparsely observed in real time, whereas control requires rapid feedback (Anand et al., 2024; Zheng et al., 2024; Boschi et al., 2026). Moreover, the safety requirements and high cost of fusion-energy systems limit the exploration of related tasks and methods in real experiments, including diagnosis and control. Therefore, simulation tools are indispensable for these tasks. A key limitation is that the computational cost of high-fidelity first-principles and integrated simulation tools limits their use within the prediction and control loops required by experiments (Fasoli et al., 2016; Agnello et al., 2024; Amorisco et al., 2024; Wang et al., 2025b). These demands have motivated efficient deep-learning models for state prediction and surrogate simulation (Morosohk et al., 2024; Wang et al., 2025a). Machine learning has already contributed to fusion research through disruption prediction (Kates-Harbeck et al., 2019; Vega et al., 2022), real-time equilibrium reconstruction and shape control (Anand et al., 2024), and feedback control of tokamak plasmas (Degrave et al., 2022; Wang et al., 2025a). Yet for a strongly nonlinear

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.

and nonstationary temporal physical system such as a tokamak fusion-energy system, an isolated one-step prediction has limited practical utility: practical workflows often require multi-step recursive rollouts to estimate future states and inform control planning (Char et al., 2024; Wang et al., 2025a).

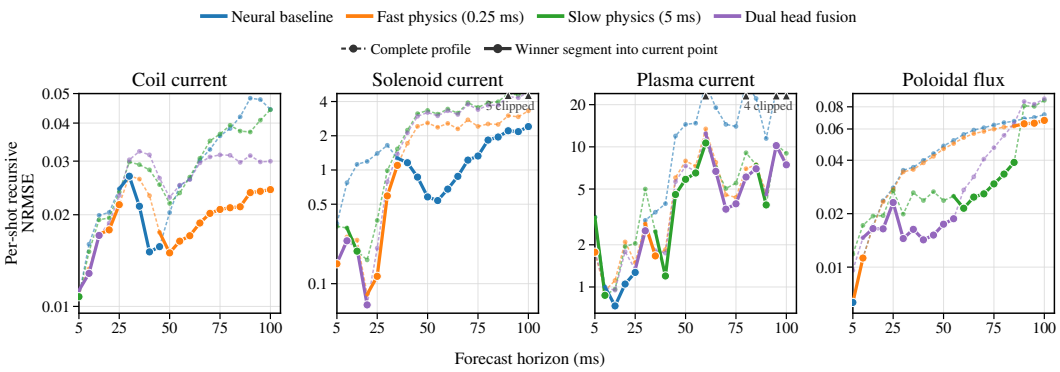

Figure 1: Per-shot NRMSE trajectories of four experts (see Appendix A.2) across physical variables for a representative sample on the real MAST device.

Multi-step recursive rollouts introduce a fundamental difficulty: each prediction is fed back as the next input, so small one-step errors can accumulate rapidly as the forecast horizon grows (Bengio et al., 2015; Ross et al., 2011; Wang et al., 2025a; Lin et al., 2026). Although various long-horizon forecasting methods have been proposed to mitigate error accumulation in general forecasting settings (Zhou et al., 2021; Nie et al., 2023; Ni et al., 2024; Zhang et al., 2025; Ortigossa et al., 2026), their suitability for multivariable fusion energy system surrogates remains unclear. Unlike conventional fluid or combustion forecasting, prediction for fusion-energy systems is an extremely multivariate, strongly coupled, and sparsely observed problem involving heterogeneous physical variables such as currents, voltages, the poloidal magnetic flux ψ, and the toroidal current density j_{tor} (Fasoli et al., 2016; Meyer et al., 2013; Char et al., 2024; Wang et al., 2025a; Boschi et al., 2026). The full plasma state cannot be measured directly, so partial observability and heterogeneous diagnostic variables further complicate prediction (Anand et al., 2024; Zheng et al., 2024; Boschi et al., 2026). In our comparisons of multiple deep-learning models (Figure 1), ***no single model is consistently optimal across all signals and rollout steps.*** These observations suggest that, at the current stage, learning long-horizon dynamics across multiple physical signals within a single model may not always yield state-of-the-art (SOTA) predictive performance. Motivated by these observations, we investigate whether multiple models (experts)—including purely data-driven models and learned models augmented with identified dynamics or simulator-derived information—can be selected adaptively to improve multistep, multivariate evolution prediction for fusion-energy systems.

To the best of our knowledge, this is the first study to systematically construct and compare multi-expert routing frameworks for multivariable recursive prediction in the fusion-energy system. We present four multi-expert routing frameworks. The first is a teacher-forced behavior-cloning Mixture-of-Experts (MoE) trained with single-step oracle expert-selection labels (Baseline-MoE). The second is a DAgger-style rollout data-aggregation MoE (DaD-MoE): the router generates deterministic recursive rollouts, experts assign hard one-step oracle labels to the learner-visited states, and the resulting labeled examples are aggregated for supervised retraining (Ross et al., 2011; Li & Zhang, 2025). The third is a reinforcement learning-based MoE optimized with proximal policy optimization (PPO-MoE), which selects experts according to errors accumulated over complete multistep rollouts (Schulman et al., 2017; Lin et al., 2026). The fourth is FIRE (Factorized Full-Information Rollout-Aware Expert Routing), a two-stage routing framework. Stage A evaluates all candidate experts for each physical head on states visited during learner rollouts, and Stage B refines the resulting router using complete-rollout errors while keeping the updated policy close to the Stage A policy.

To evaluate the effectiveness of the above methods, we conduct experiments in two representative fusion-energy settings. First, we conduct the validation of multi-expert routing on tokamak simulation setting. This setting involves complex vertical displacement event (VDE) dynamics and requires tracking additional plasma-geometry variables, including the plasma vertical position and X-point locations, which we also evaluate here (Lvovskiy et al., 2025). PPO-MoE achieves the lowest

complete-rollout error in this setting, reducing the error by 6.38% relative to Baseline-MoE. Building on this simulation-based validation, we apply the same framework to real-world experimental measurement from the MAST device (Meyer et al., 2013; Boschi et al., 2026). The experimental trajectories additionally contain diagnostic noise, heterogeneous measurements, shot-to-shot operating variation, and device-specific effects that are simplified or absent in the simulator, making the real-world device evaluation a more demanding test of deployment robustness. FIRE establishes a favorable accuracy: its complete-rollout error is 12.29% lower than that of the best fixed-expert reference and 7.81% lower than Baseline-MoE. Together, the two experiments demonstrate the potential of multi-expert routing to improve recursive prediction in complex fusion-energy systems and delineate the conditions under which routing policies require revalidation as geometry, simulators, temporal scales, or expert composition change.

Our contributions are threefold:

- We evaluate how heterogeneous models perform across physical variables and rollout stages. In the studied expert bank, no single model dominates across all signals and time steps, and teacher-forced one-step rankings can disagree with rankings induced by complete recursive rollouts.
- To the best of our knowledge, this is the first systematic construction and comparison of DaD-MoE, PPO-MoE, and FIRE for multivariable recursive trajectory prediction in the fusion-energy system.
- We evaluate the three routing frameworks on experimental data from the real-world MAST device. All three routing frameworks outperform the prescribed baseline conditions in complete-rollout error. FIRE further reduces complete-rollout error by 12.29% relative to the best fixed-expert reference and by 7.81% relative to Baseline-MoE.

2 Related Work

Deep learning in fusion energy. Deep learning has been applied to tokamak profile prediction, transport modeling, disruption prediction, equilibrium reconstruction, and plasma control (Abbate et al., 2021; Morosohk et al., 2024; Wang et al., 2025b; Kates-Harbeck et al., 2019; Anand et al., 2024; Degrave et al., 2022). Work on multivariable experimental trajectories primarily develops fast surrogate models for control and trajectory optimization (Seo et al., 2021; 2022; Char et al., 2023). Recent studies address direct trajectory forecasting, including full-shot recurrent prediction on DIII-D and multimodal MAST diagnostic forecasting (Char et al., 2024; Boschi et al., 2026). Despite this progress, long-horizon prediction of coupled plasma variables remains challenging: different variables exhibit heterogeneous temporal dynamics, and recursive errors in one variable can propagate through the coupled system. We therefore formulate tokamak forecasting as a multivariate learned-dynamics problem, with an emphasis on adaptive composition of heterogeneous predictors over long recursive rollouts.

Mixtures of experts for heterogeneous time-series dynamics. Mixture-of-experts (MoE) routing provides conditional computation for heterogeneous predictors (Jacobs et al., 1991; Shazeer et al., 2017). Recent time-series methods learn routers over specialized experts or temporal scales, while online variants adapt expert mixtures from sequential feedback and learned-dynamics methods select among prediction heads (Ni et al., 2024; Zhang et al., 2025; Ortigossa et al., 2026; Saqur et al., 2025; Zhao et al., 2025; Seo et al., 2020). Our setting instead routes among a frozen bank of heterogeneous physical predictors. Decisions span coupled recursive states and the objective is rollout accuracy rather than only local prediction error. Detailed comparisons are provided in Appendix A.8.

Sequential routing under recursive distribution shift. Recursive routing is sequential: each expert choice changes the next state and hence later observations, creating the distribution shift studied in sequential prediction and imitation learning. Scheduled sampling, DAgger, Data as Demonstrator, and interactive feedback methods address learner-induced states (Bengio et al., 2015; Ross et al., 2011; Venkatraman et al., 2015; Li & Zhang, 2025); AggreVaTe and policy-optimization methods incorporate longer-horizon objectives (Ross & Bagnell, 2014; Sun et al., 2017; Peters et al., 2010; Schulman et al., 2015; 2017; Abdolmaleki et al., 2018); and model-based offline reinforcement learning emphasizes complete-rollout evaluation under compounding error (Lin et al., 2026). We therefore cast routing as sequential decision-making over frozen models and evaluate the multivariable forecasts over recursive rollouts. Detailed discussion is provided in Appendix A.8.

3 METHODOLOGY

To improve long-horizon rollout stability and mitigate the accumulation of prediction errors in multivariable physical prediction, we construct four routing frameworks with complementary training signals: Baseline-MoE learns one-step expert-selection labels on ground-truth trajectories; DaD-MoE extends this supervision to learner-induced states; PPO-MoE optimizes expert selection from complete-trajectory returns; and FIRE combines full-information per-head supervision with rollout refinement. Figure 2 provides a visual overview of the route-specific router architectures and data flow, covering input encoding, expert-context fusion, and route-specific expert selection. The corresponding training inputs and objectives are summarized in Appendix A.3.

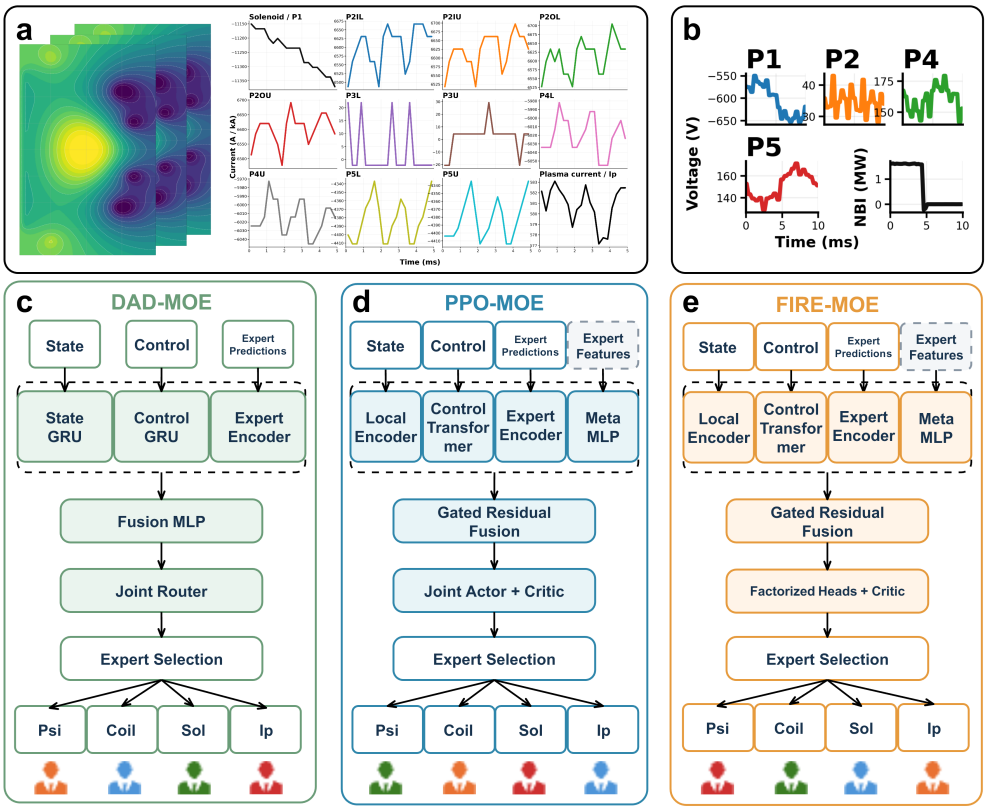

Figure 2: Overview of tokamak forecasting inputs and mixture-of-experts architectures. (a) Poloidal magnetic flux (ψ) and current signals from the plasma and poloidal field coils; (b) five control variables; (c–e) model architectures and data flows of DaD-MoE, PPO-MoE, and FIRE, respectively.

3.1 RECURSIVE PREDICTION AND ROUTING WITH A FROZEN EXPERT BANK

At each forecast step t, let x_t and $\widehat{x}_t$ denote the ground-truth and recursively predicted multivariable physical states, respectively, with $t = 0, \ldots, T$. We partition the state into H semantically coherent output heads,

$$x_t = (x_{t,1}, \ldots, x_{t,H}), \qquad x_{t,h} \in \mathbb{R}^{d_h}. \tag{1}$$

For each head h, a frozen expert bank provides K_h candidate predictors $f_{h,e}$, where $e \in \mathcal{E}_h = \{1, \ldots, K_h\}$. A joint action $a_t = (a_{t,1}, \ldots, a_{t,H})$ selects one candidate for each head from $\mathcal{A} = \prod_{h=1}^{H} \mathcal{E}_h$. Baseline-MoE, DaD-MoE, and PPO-MoE use a flat categorical policy over $\mathcal{A}$, whereas FIRE factorizes the policy across heads. Starting from the recorded initial state, the selected frozen predictors recursively advance the complete state:

$$\widehat{x}_0 = x_0, \qquad \widehat{x}_{t+1,h} = f_{h,a_{t,h}}(\widehat{x}_t, u_t), \qquad t = 0, \ldots, T-1. \tag{2}$$

All expert parameters remain frozen, so each routing action changes both the next prediction and the states encountered at later steps. At deployment, each route selects the argmax action in its implemented action space.

The router observes

$$o_t = g(\widehat{x}_{\le t}, u_{\ge t}, p, \xi_t), \tag{3}$$

where u_t denotes the current control input, $u_{\ge t}$ the permitted control sequence from step t onward, p the position features, and ξ_t the expert-derived context. We evaluate a routing policy π by its complete recursive trajectory,

$$J(\pi) = \mathbb{E}_\tau[\mathcal{E}_{\text{roll}}(\widehat{x}_{1:T}(\pi), x_{1:T})], \tag{4}$$

where $\mathbb{E}_\tau$ averages over trajectories or shots and $\mathcal{E}_{\text{roll}}$ is the domain-specific complete-rollout error. Unlike a one-step oracle, $J(\pi)$ accounts for the future states induced by earlier routing decisions. Additional expert-bank, observation, and implementation details are provided in Appendix A.2, A.3.

3.2 Baseline-MoE: teacher-forced behavior cloning

For a teacher-forced state $\widehat{x}_t$, current control u_t, and ground-truth next state x_{t+1}, the normalized one-step cost and per-head oracle are

$$\begin{aligned} c_{h,e}(\widehat{x}_t, x_{t+1}) &= \frac{1}{d_h}\sum_{j=1}^{d_h}\left(\frac{f_{h,e}(\widehat{x}_t, u_t)_j - x_{t+1,h,j}}{\sigma_{h,j}}\right)^2, \\ a_h^*(\widehat{x}_t, x_{t+1}) &= \arg\min_{e\in\mathcal{E}_h} c_{h,e}(\widehat{x}_t, x_{t+1}), \end{aligned} \tag{5}$$

where j indexes coordinates within head h and $\sigma_{h,j} > 0$ is a fixed training-split scale. Because the joint cost is additive, $C(a; \widehat{x}_t, x_{t+1}) = \sum_{h=1}^{H} c_{h,a_h}(\widehat{x}_t, x_{t+1})$ for $a \in \mathcal{A}$, and the per-head minimizers form the hard joint oracle $a^* = (a_1^*, \dots, a_H^*)$.

Under teacher forcing, $\widehat{x}_{\le t} = x_{\le t}$. Baseline-MoE therefore forms $D_0 = \{(o_t^{\text{TF}}, a_t^*)\}$ from all valid router-training steps, with $o_t^{\text{TF}} = g(x_{\le t}, u_{\ge t}, p, \xi_t)$ and $a_t^* = a^*(x_t, x_{t+1})$, and minimizes

$$\mathcal{L}_{\text{BC}}(\theta; D_0) = -\mathbb{E}_{(o,a^*)\sim D_0} \log \pi_\theta(a^* \mid o), \tag{6}$$

where θ denotes the router parameters and π_θ is the flat joint categorical policy over $\mathcal{A}$. At deployment, the router receives recursively predicted states rather than teacher-forced histories, creating the state-distribution shift addressed by DaD-MoE. Further normalization and tie-breaking details are given in Appendix A.3.

3.3 DaD-MoE: hard-label aggregation on learner states

Let $\pi_0 = \pi_{\text{BC}}$ denote the Baseline-MoE router. For aggregation rounds $k = 1, \dots, K_{\text{DaD}}$, where K_{DaD} is the maximum number of rounds, π_{k-1} generates a deterministic recursive rollout on each router-training trajectory by selecting

$$a_t^{(k)} = \arg\max_{a\in\mathcal{A}} \pi_{k-1}(a \mid o_t^{(k)}).$$

The learner-selected action drives Eq. 2, so each visited state is predicted by the current router rather than by the oracle. At these states, the one-step oracle is used only to construct hard labels:

$$\begin{aligned} o_t^{(k)} &= g(\widehat{x}_{\le t}^{(k)}, u_{\ge t}, p, \xi_t^{(k)}), \\ a_t^{*(k)} &= a^*(\widehat{x}_t^{(k)}, x_{t+1}). \end{aligned} \tag{7}$$

The ground-truth state is therefore used for labeling, but is not supplied to the router during the learner rollout.

Let $\mathcal{D}_{\text{router}}^{\text{train}}$ denote the router-training trajectories and T_τ the number of valid forecast steps in trajectory τ. The round-k samples and aggregated training set are

$$\begin{aligned} R_k &= \left\{ \left(o_t^{(k)}, a_t^{*(k)}\right) \,\middle|\, \tau \in \mathcal{D}_{\text{router}}^{\text{train}},\ t = 0, \dots, T_\tau - 1 \right\}, \\ D_k &= D_0 \cup \bigcup_{i=1}^{k} R_i. \end{aligned} \tag{8}$$

The router is warm-started from π_{k-1} and trained on D_k with the hard-label loss in Eq. 6. Teacher-forced validation cross-entropy selects the checkpoint within each round, while complete-rollout validation error selects among completed rounds. DaD-MoE thus adapts the training-state distribution to learner-induced states while retaining Baseline-MoE's one-step oracle and supervised objective. Further aggregation and selection details are given in Appendix A.3.

3.4 PPO-MoE: COMPLETE-ROLLOUT RETURN OPTIMIZATION

PPO-MoE formulates router training as a finite-horizon Markov decision process. The state contains the current predicted state, permitted future controls, optional position features, and candidate-expert context; each action selects one frozen expert for every output head. Episodes follow the deterministic transition in Eq. 2, with sampled actions during training and argmax actions at deployment. Ground-truth states are used only to compute training potentials and the candidate-cost auxiliary. The actor and shared encoder are initialized from the paired Baseline-MoE checkpoint, and the critic is pretrained on complete-episode returns.

A training potential Φ_t combines training-split-normalized prefix errors over all transitions with errors from the final 20% of each episode, weighted by 0.8 and 0.2. We define

$$r_t = -(\Phi_{t+1} - \Phi_t), \qquad G_t = \sum_{\ell=t}^{T-1} \gamma^{\ell-t} r_\ell, \qquad \Phi_0 = 0, \qquad \gamma = 1. \tag{9}$$

Thus, G_t is the complete-episode return and $G_0 = -\Phi_T$. For rollout batches, we optimize the standard clipped PPO objective with value and entropy regularization, together with an early candidate-cost auxiliary:

$$\mathcal{L}_{\mathrm{PPO}} = \mathcal{L}_{\mathrm{clip}} + \lambda_V \mathcal{L}_V - \lambda_H \mathcal{H}(\pi_\theta) + \alpha_{\mathrm{aux}}(m)\mathcal{L}_{\mathrm{aux}}. \tag{10}$$

The auxiliary coefficient decays linearly to zero during the first 20% of PPO updates. The detail of implementation is introduced in Appendix A.3.3.

3.5 FIRE: FULL-INFORMATION SUPERVISION AND ROLLOUT REFINEMENT

FIRE combines full-information supervision on learner states with Stage-A-anchored rollout refinement. Its policy factorizes across output heads:

$$\pi_\theta(a \mid o) = \prod_{h=1}^{H} \pi_{\theta,h}(a_h \mid o). \tag{11}$$

This representation emits $\sum_{h=1}^{H} K_h$ logits instead of $\prod_{h=1}^{H} K_h$ joint logits, while the recursive predictor still advances all output heads jointly through Eq. 2. FIRE is initialized from the exact per-head marginals of the joint behavior-cloned policy; the construction and residual-head initialization are given in Appendix A.3.

Stage A: full-information supervision on learner states. At round k, the factorized parent policy π_k generates deterministic learner rollouts. For every visited observation o and head h, FIRE retains the costs of all candidate experts rather than only the minimum-cost label. Let

$$\delta_{h,e}(o) = c_{h,e}(\widehat{x}_t, x_{t+1}) - \min_{j \in \mathcal{E}_h} c_{h,j}(\widehat{x}_t, x_{t+1})$$

be the candidate regret, and let $s_h(o) > 0$ be its scale. FIRE forms the prior-weighted mirror target

$$q_{k,h}(e \mid o) = \frac{\mathrm{stopgrad}[\pi_{k,h}(e \mid o)] \exp[-\delta_{h,e}(o)/s_h(o)]}{\sum_{j \in \mathcal{E}_h} \mathrm{stopgrad}[\pi_{k,h}(j \mid o)] \exp[-\delta_{h,j}(o)/s_h(o)]}. \tag{12}$$

Here $\mathrm{stopgrad}$ fixes the parent-policy probabilities during target construction, and $s_h(o)$ uses the median positive regret with standard-deviation and unit-scale fallbacks. Let $\mathcal{D}_{\mathrm{A}}^{(k)}$ contain the teacher-forced and learner-state observations accumulated through round k. The Stage-A student minimizes

$$\mathcal{L}_A = -\mathbb{E}_{o \sim \mathcal{D}_{\mathrm{A}}^{(k)}} \left[\frac{1}{H} \sum_{h=1}^{H} \sum_{e \in \mathcal{E}_h} q_{k,h}(e \mid o) \log \pi_{\theta,h}(e \mid o) \right]. \tag{13}$$

Thus, unlike a hard DaD label, $q_{k,h}$ preserves relative candidate quality and near ties without enumerating the Cartesian action space.

Stage B: trajectory refinement with Stage-A anchoring. Let π_A denote the selected Stage-A policy. Stage B initializes its actor from π_A, fits a critic to complete-episode returns, and applies the PPO terms in Eq. 10 with the immediate-cost auxiliary disabled. It anchors the refined policy to π_A through the normalized per-head KL loss

$$\mathcal{L}_{\text{anchor}} = \mathbb{E}_{o\sim\mathcal{D}_{\text{B}}}\left[\frac{\sum_{h=1}^{H} D_{\text{KL}}(\pi_{\theta,h}(\cdot \mid o) \,\|\, \pi_{A,h}(\cdot \mid o))}{\log|\mathcal{A}|}\right], \tag{14}$$

where $\mathcal{D}_{\text{B}}$ denotes Stage-B rollout observations and $|\mathcal{A}| = \prod_{h=1}^{H} K_h$. The Stage-B objective is

$$\mathcal{L}_B = \mathcal{L}_{\text{clip}} + \lambda_V \mathcal{L}_V - \lambda_H \mathcal{H}(\pi_\theta) + \beta \mathcal{L}_{\text{anchor}}, \tag{15}$$

where $\beta \geq 0$ is the Stage-A anchoring coefficient. FIRE therefore combines full-information candidate supervision with complete-rollout refinement while keeping the refined policy close to the Stage-A solution. More implementation details are provided in Appendices A.3 and A.3.3.

4 Experiments

4.1 Experimental Setup

Dataset. We evaluate the routing methods on two representative tokamak datasets, one from a simulated fusion-energy system (MAST-U) and the other from a real-world fusion-energy system (MAST). For MAST-U, we generate our own dataset by simulating vertical-displacement events with FreeGSNKE (Amorisco et al., 2024), following the MAST-U control setting described in prior work (Lvovskiy et al., 2025). The dataset contains 4,000 expert-training, 2,400 router-training, 800 validation, and 800 formal-test trajectories; each trajectory is evaluated over 20 recursive steps at 0.5 ms per step. The MAST dataset contains experimental trajectories from the real MAST device (Meyer et al., 2013; Anand et al., 2024; Boschi et al., 2026). Its strict split contains 3,321 expert-training, 1,992 router-training, 664 validation, and 664 formal-test shots, with a common 367-shot cohort for the 100-ms endpoint. Splits are disjoint by trajectory or shot identity, and the formal test set is reserved for final evaluation. Detailed descriptions of data generation, preprocessing, variable definitions, normalization, and geometry variables are provided in Appendix A.1, A.4.3.

Expert banks. Each dataset uses a fixed bank of physical predictors, with all expert parameters frozen throughout router training and evaluation. For the simulated fusion-energy system (MAST-U), we construct three expert models: TFLinearFusion, SINDyC-dual, and TF-NN. For the real-world fusion-energy system (MAST), we construct four expert models: Baseline, ODE0.25, ODE5, and Dual. Detailed expert definitions and provenance are provided in Appendix A.2.

4.2 Results

4.2.1 Simulated Fusion-Energy System

We evaluate the routing methods on simulated MAST-U vertical-displacement-event trajectories. The comparison uses the frozen three-expert bank and probability-ensemble aggregation, which averages router probabilities before selecting actions. FIRE is evaluated with $\beta = 0.1$ in the Stage-B objective, where β is the anchoring weight in Eq. 15.

Table 1 reports the comparison at the 10-ms endpoint. DaD-MoE, PPO-MoE, and FIRE-MoE all outperform Baseline-MoE. PPO-MoE achieves the lowest error across all five physical variables and the lowest Macro relative-ℓ_2 error, 4.8638×10^{-2}. It outperforms the strongest fixed expert, reducing Macro error by **15.46%** relative to Baseline-MoE. FIRE-MoE and DaD-MoE reduce Macro error by **14.38%** and **12.62%**, respectively. Figure 3 shows a representative comparison of ψ-field errors.

To examine routing behavior and the relationship between one-step and recursive prediction, Appendices A.6 and A.4.2 reports validation-set routing-probability entropy, together with one-step and full-rollout physical errors, representative trajectories, and field-error maps. Together, these results demonstrate complementary strengths across the experts and support the central observation that one-step accuracy alone does not determine recursive performance rankings. We therefore use complete-rollout error as the primary metric for comparing routing methods.

Table 1: MAST-U physical errors at the 10-ms endpoint (the twentieth 0.5-ms step) under probability-ensemble aggregation. Bold entries are the lowest errors in their respective columns.

Model	Active	Passive	I_p	ψ	J_{tor}	Macro
TFLinearFusion	0.0041740	0.24232	0.0029333	0.0078057	0.018000	0.055046
	(+7.09%)	(+4.09%)	(+5.28%)	(+5.98%)	(+5.80%)	(+4.32%)
SINDyC-dual	0.0044901	0.24890	0.0031433	0.0082440	0.019212	0.056799
	(+0.06%)	(+1.49%)	(-1.50%)	(-0.70%)	(-0.54%)	(-1.28%)
TF-NN	0.0040828	0.24094	0.0030025	0.0079794	0.019193	0.055039
	(+9.13%)	(+4.64%)	(+3.05%)	(+3.89%)	(-0.44%)	(+4.34%)
Baseline-MoE	0.0044927	0.25267	0.0030968	0.0083020	0.019109	0.057533
	(+0.00%)	(+0.00%)	(+0.00%)	(+0.00%)	(+0.00%)	(+0.00%)
PPO-MoE	**0.0038585**	**0.21208**	**0.0028086**	**0.0072648**	**0.017184**	**0.048638**
	(+14.12%)	**(+16.06%)**	**(+9.31%)**	**(+12.49%)**	**(+10.07%)**	**(+15.46%)**
DaD-MoE	0.0039955	0.21936	0.0028372	0.0075294	0.017627	0.050270
	(+11.07%)	(+13.18%)	(+8.38%)	(+9.31%)	(+7.75%)	(+12.62%)
FIRE ($\beta = 0.1$)	0.0039367	0.21493	0.0028208	0.0073488	0.017260	0.049259
	(+12.38%)	(+14.94%)	(+8.91%)	(+11.48%)	(+9.67%)	(+14.38%)

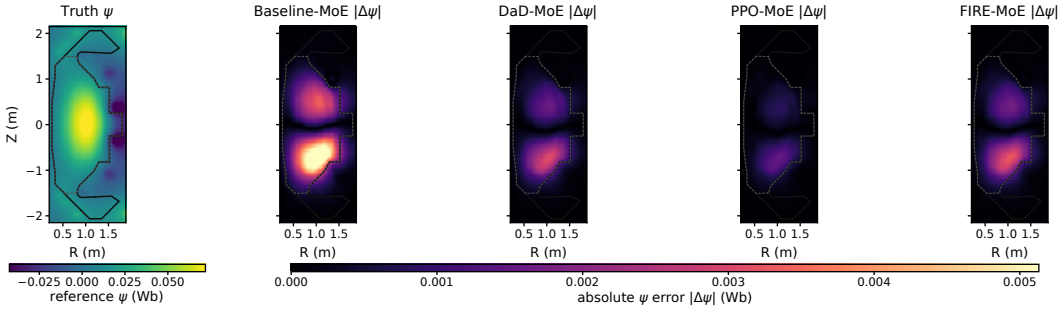

Figure 3: MAST-U ψ ground-truth and absolute-error maps at the 10-ms endpoint.

Because VDE dynamics are sensitive to small boundary and X-point displacements, we evaluate seven geometry variables through recursive replay. At 10 ms, endpoint position errors across the routing methods range from 1.62–3.32 cm, compared with 1.78–3.68 cm for Baseline-MoE. **These position errors are on the same centimetre scale as the 2–10 cm range reported in MAST-U shape-control studies** (Anand et al., 2024; Lvovskiy et al., 2025). Detailed results for these geometry variables are provided in Appendix A.4.3.

4.2.2 Real-World Fusion-Energy System

We evaluate the routing methods on experimental trajectories from the real MAST device. The formal comparison uses the frozen four-expert bank under the same probability-ensemble protocol. FIRE is evaluated with $\beta = 0.05$ in the Stage-B objective.

Table 2 reports the formal full-rollout comparison under probability-ensemble aggregation. FIRE-MoE with $\boldsymbol{\beta = 0.05}$ achieves the lowest pooled Macro NRMSE of $\mathbf{0.06506154}$, corresponding to a **7.81%** reduction relative to Baseline-MoE and a **12.29%** reduction relative to the strongest fixed expert, Dual ($\mathbf{0.07417391}$). DaD-MoE ranks second with a pooled Macro NRMSE of 0.06550750, improving over Dual by 11.68%. Among the router methods, FIRE-MoE achieves the lowest error for Coil and I_p, whereas DaD-MoE achieves the lowest error for Solenoid and ψ. PPO-MoE and Baseline-MoE also improve over Dual by 8.31% and 4.86%, respectively. **Thus, all router methods outperform the strongest fixed expert in pooled Macro NRMSE, with FIRE-MoE providing the best overall result and DaD-MoE the second-best.** Figure 4 visualizes the ψ-field errors of the four MoE routers for one 100-ms formal-test shot. The one-step and 100-ms endpoint results are reported in Appendix A.4.1. **Together with the full-rollout comparison above, these results provide**

empirical evidence that adaptive routing can improve recursive multivariable forecasting in a real-world fusion-energy system.

Table 2: Formal MAST full recursive-rollout comparison under probability-ensemble aggregation. Bold entries are the lowest errors in their respective columns. The superscript * marks the strongest fixed expert according to pooled Macro NRMSE.

Model	Coil	Solenoid	I_p	ψ	Macro
Baseline expert	0.11306541	0.12192044	0.06325351	0.08683299	0.09626809
	(-26.59%)	(-98.94%)	(-13.62%)	(-14.23%)	(-36.41%)
ODE0.25 expert	0.08925495	0.06655452	0.05744676	0.08402911	0.07432134
	(+0.07%)	(-8.60%)	(-3.19%)	(-10.54%)	(-5.31%)
ODE5 expert	0.09105119	0.06310403	0.05613676	0.11173256	0.08050614
	(-1.95%)	(-2.97%)	(-0.84%)	(-46.98%)	(-14.08%)
Dual expert*	0.08772451	0.07162403	0.05821094	0.07913616	0.07417391
	(+1.78%)	(-16.87%)	(-4.57%)	(-4.10%)	(-5.10%)
Baseline-MoE	0.08931343	0.06128494	0.05566874	0.07601877	0.07057147
	(+0.00%)	(+0.00%)	(+0.00%)	(+0.00%)	(+0.00%)
PPO-MoE	0.08751399	0.05751838	0.05283465	0.07418335	0.06801259
	(+2.01%)	(+6.15%)	(+5.09%)	(+2.41%)	(+3.63%)
DaD-MoE	0.08833368	**0.05493872**	0.04789438	**0.07086322**	0.06550750
	(+1.10%)	**(+10.36%)**	(+13.97%)	**(+6.78%)**	(+7.18%)
FIRE ($\beta = 0.05$)	**0.08422290**	0.05676223	**0.04713915**	0.07212188	**0.06506154**
	(+5.70%)	(+7.38%)	**(+15.32%)**	(+5.13%)	**(+7.81%)**

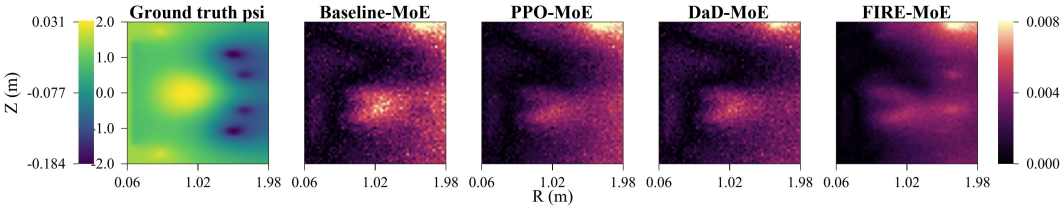

Figure 4: MAST ψ ground-truth and absolute-error maps at 100 ms (formal-test shot 16618)

To further understand how routing decisions evolve and what factors influence recursive prediction, we analyze FIRE mechanisms, validation-only routing-probability entropy, and the dependence of routing performance on expert banks and training data. These supplementary analyses include FIRE component ablations and reference-weight robustness, routing specialization during recursive rollouts, and expert-bank transfer and router data exposure studies. The corresponding analyses are reported in Appendices A.5, A.6, and A.7, respectively.

5 Conclusion

We study adaptive expert routing for multistep, multivariable tokamak forecasting with a frozen bank of heterogeneous predictors. We compare Baseline-MoE, DaD-MoE, PPO-MoE, and FIRE under recursive evaluation on simulated MAST-U vertical-displacement-event trajectories and trajectories from the real-world MAST fusion-energy system. Complete-rollout rankings do not follow teacher-forced one-step accuracy. Under probability-ensemble aggregation, PPO-MoE achieves the lowest MAST-U complete-rollout error, improving over Baseline-MoE by 6.38%. On trajectories from the real-world MAST fusion-energy system, FIRE achieves the lowest complete-rollout Macro NRMSE, improving over the strongest fixed expert and Baseline-MoE by 12.29% and 7.81%, respectively. Together, these results show that adaptive MoE routing improves recursive multivariable forecasting by exploiting complementary experts across physical variables and recursive horizons, while complete-rollout evaluation provides an appropriate criterion for comparing routing policies.

AI USE STATEMENT

The authors used AI tools to help polish the manuscript text and to inspire the design of the flowchart.

REFERENCES

J. Abbate, R. Conlin, and E. Kolemen. Data-driven profile prediction for DIII-D. *Nuclear Fusion*, 61 (4):046027, 2021. doi: 10.1088/1741-4326/abe08d. URL https://doi.org/10.1088/1741-4326/abe08d.

Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Remi Munos, Nicolas Heess, and Martin Riedmiller. Maximum a posteriori policy optimisation. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=S1ANxQW0b.

A. Agnello, N. C. Amorisco, A. Keats, G. K. Holt, J. Buchanan, S. Pamela, C. Vincent, and G. McArdle. Emulation techniques for scenario and classical control design of tokamak plasmas. *Physics of Plasmas*, 31(4):043091, 2024. doi: 10.1063/5.0187822.

N. C. Amorisco, A. Agnello, G. Holt, M. Mars, J. Buchanan, and S. Pamela. FreeGSNKE: A python-based dynamic free-boundary toroidal plasma equilibrium solver. *Physics of Plasmas*, 31 (4):042517, 2024. doi: 10.1063/5.0188467.

H. Anand, W. Wehner, D. Eldon, et al. Real-time plasma equilibrium reconstruction and shape control for the MAST Upgrade tokamak. *Nuclear Fusion*, 64(8):086051, 2024. doi: 10.1088/1741-4326/ad5c80.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, volume 28, 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/hash/e995f98d56967d946471af29d7bf99f1-Abstract.html.

Tobia Boschi, Andrea Loreti, Nicola C. Amorisco, Rodrigo H. Ordonez-Hurtado, Cécile Rousseau, George K. Holt, Eszter Székely, Alexander Whittle, Samuel Jackson, Adriano Agnello, Stanislas Pamela, Alessandra Pascale, Robert Akers, Juan Bernabe Moreno, Sue Thorne, and Mykhaylo Zayats. TokaMind: A multi-modal transformer foundation model for tokamak plasma dynamics. *arXiv preprint arXiv:2602.15084*, 2026. URL https://arxiv.org/abs/2602.15084.

Ian Char, Joseph Abbate, László Bardóczi, Mark D. Boyer, Youngseog Chung, Rory Conlin, Keith Erickson, Viraj Mehta, Nathan Richner, Egemen Kolemen, and Jeff Schneider. Offline model-based reinforcement learning for tokamak control. In *Proceedings of the 5th Conference on Learning for Dynamics and Control*, volume 211 of *Proceedings of Machine Learning Research*, pp. 1357–1372. PMLR, 2023. URL https://proceedings.mlr.press/v211/char23a.html.

Ian Char, Youngseog Chung, Joseph Abbate, Egemen Kolemen, and Jeff Schneider. Full shot predictions for the DIII-D tokamak via deep recurrent networks. *arXiv preprint arXiv:2404.12416*, 2024. doi: 10.48550/arXiv.2404.12416. URL https://arxiv.org/abs/2404.12416.

Jonas Degrave, Federico Felici, Jonas Buchli, et al. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602:414–419, 2022. doi: 10.1038/s41586-021-04301-9.

A. J. H. Donné. The european roadmap towards fusion electricity. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 377(2141):20170432, 2019. doi: 10.1098/rsta.2017.0432.

A. Fasoli, S. Brunner, W. A. Cooper, J. P. Graves, P. Ricci, O. Sauter, and L. Villard. Computational challenges in magnetic-confinement fusion physics. *Nature Physics*, 12:411–423, 2016. doi: 10.1038/nphys3744.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991. doi: 10.1162/neco.1991.3.1.79.

Julian Kates-Harbeck, Alexey Svyatkovskiy, and William Tang. Predicting disruptions in tokamaks using deep learning. *Nature*, 568:526–531, 2019. doi: 10.1038/s41586-019-1116-4.

Yichen Li and Chicheng Zhang. Interactive and hybrid imitation learning: Provably beating behavior cloning. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-1590. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/4450fb861dfe8e9e3521a26aa6ab4c43-Abstract-Conference.html.

Haoxin Lin, Siyuan Xiao, Yi-Chen Li, Zhilong Zhang, Yihao Sun, Chengxing Jia, and Yang Yu. Adm-v2: Pursuing full-horizon roll-out in dynamics models for offline policy learning and evaluation. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/1651a9bfa452a13c9f26f71bfef1f68c-Abstract-Conference.html.

Andrey Lvovskiy, Himank Anand, Anders S. Welander, Martin Kochan, Charles Vincent, Graham McArdle, Jack J. Lovell, Z. A. Xing, Jayson L. Barr, E. Cho, Brian Sammuli, D. A. Humphreys, N. W. Eidietis, V. Soukhanovskii, A. W. Leonard, A. O. Nelson, Andrew Thornton, and James Harrison. Framework for assessment of magnetic equilibrium controller performance on the MAST upgrade spherical tokamak. *Plasma Physics and Controlled Fusion*, 67(7):075003, 2025. doi: 10.1088/1361-6587/ade3fd.

H. Meyer et al. Overview of physics results from MAST towards ITER/DEMO and the MAST upgrade. *Nuclear Fusion*, 53(10):104008, 2013. doi: 10.1088/0029-5515/53/10/104008.

Shira Morosohk, Zibo Wang, Sai Tej Paruchuri, Tariq Rafiq, and Eugenio Schuster. Optimal control of the electron temperature profile in DIII-D using machine learning surrogate models. *Fusion Engineering and Design*, 206:114615, 2024. doi: 10.1016/j.fusengdes.2024.114615.

Ronghao Ni, Zinan Lin, Shuaiqi Wang, and Giulia Fanti. Mixture-of-linear-experts for long-term time series forecasting. In *Proceedings of the 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pp. 4672–4680. PMLR, 2024. URL https://proceedings.mlr.press/v238/ni24a.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. URL https://arxiv.org/abs/2211.14730.

Jef Ongena, R. Koch, R. Wolf, and H. Zohm. Magnetic-confinement fusion. *Nature Physics*, 12: 398–410, 2016. doi: 10.1038/nphys3745.

Evandro S. Ortigossa, Guy Lutsker, and Eran Segal. Mohets: Long-term time series forecasting with mixture-of-heterogeneous-experts. *arXiv preprint arXiv:2601.21866*, 2026. URL https://arxiv.org/abs/2601.21866.

Jan Peters, Katharina Mülling, and Yasemin Altun. Relative entropy policy search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 24, pp. 1607–1612, 2010. doi: 10.1609/aaai.v24i1.7727. URL https://ojs.aaai.org/index.php/AAAI/article/view/7727.

Stephane Ross and J. Andrew Bagnell. Reinforcement and imitation learning via interactive no-regret learning. *arXiv preprint arXiv:1406.5979*, 2014. doi: 10.48550/arXiv.1406.5979.

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

Raeid Saqur, Anastasis Kratsios, Florian Krach, Yannick Limmer, Jacob-Junqi Tian, John Willes, Blanka Horvath, and Frank Rudzicz. Filtered not mixed: Filtering-based online gating for mixture of large language models. In *International Conference on Learning Representations*, 2025. URL

https://proceedings.iclr.cc/paper_files/paper/2025/file/d4c2f25bf0c33065b7d4fb9be2a9add1-Paper-Conference.pdf.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1889–1897. PMLR, 2015. URL https://proceedings.mlr.press/v37/schulman15.html.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. doi: 10.48550/arXiv.1707.06347.

Jaemin Seo, Y. S. Na, Boseong Kim, Chan Young Lee, Min Seok Park, Seong Jun Park, and Young Hwan Lee. Feedforward beta control in the KSTAR tokamak by deep reinforcement learning. *Nuclear Fusion*, 61(10):106010, 2021. doi: 10.1088/1741-4326/ac121b. URL https://doi.org/10.1088/1741-4326/ac121b.

Jaemin Seo, Y. S. Na, Boseong Kim, Chan Young Lee, Min Seok Park, Seong Jun Park, and Young Hwan Lee. Development of an operation trajectory design algorithm for control of multiple 0D parameters using deep reinforcement learning in KSTAR. *Nuclear Fusion*, 62(8):086049, 2022. doi: 10.1088/1741-4326/ac79be. URL https://doi.org/10.1088/1741-4326/ac79be.

Younggyo Seo, Kimin Lee, Ignasi Clavera, Thanard Kurutach, Jinwoo Shin, and Pieter Abbeel. Trajectory-wise multiple choice learning for dynamics generalization in reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://papers.neurips.cc/paper_files/paper/2020/file/9739efc4f01292e764c86caa59af353e-Paper.pdf.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=B1ckMDqlg.

Wen Sun, Arun Venkatraman, Geoffrey J. Gordon, Byron Boots, and J. Andrew Bagnell. Deeply AggreVaTeD: Differentiable imitation learning for sequential prediction. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3309–3318. PMLR, 2017. URL https://proceedings.mlr.press/v70/sun17d.html.

J. Vega, A. Murari, S. Dormido-Canto, G. Ratta, M. Gelfusa, and JET Contributors. Disruption prediction with artificial intelligence techniques in tokamak plasmas. *Nature Physics*, 18:741–750, 2022. doi: 10.1038/s41567-022-01602-2.

Arun Venkatraman, Martial Hebert, and J. Andrew Bagnell. Improving multi-step prediction of learned time series models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015. doi: 10.1609/aaai.v29i1.9590. URL https://ojs.aaai.org/index.php/AAAI/article/view/9590.

Allen M. Wang, Alessandro Pau, Cristina Rea, Oswin So, Charles Dawson, Olivier Sauter, Mark D. Boyer, Anna Vu, Cristian Galperti, Chuchu Fan, Antoine Merle, Yoeri Poels, Cristina Venturini, Federico Felici, Stefano Marchioni, and TCV Team. Learning plasma dynamics and robust rampdown trajectories with predict-first experiments at TCV. *Nature Communications*, 16:8877, 2025a. doi: 10.1038/s41467-025-63917-x. URL https://www.nature.com/articles/s41467-025-63917-x.

Zibo Wang, Eugenio Schuster, Tariq Rafiq, Y. Huang, Z. Luo, Q. Yuan, and Jayson Lee Barr. Enabling model-based scenario control in EAST by fast surrogate modeling within COTSIM. *Fusion Engineering and Design*, 215:114969, 2025b. doi: 10.1016/j.fusengdes.2025.114969.

Huanyao Zhang, Jiaye Lin, Wentao Zhang, Haitao Yuan, and Guoliang Li. Adamixt: Adaptive weighted mixture of multi-scale expert transformers for time series forecasting. In *Proceedings of*

the Thirty-Fourth International Joint Conference on Artificial Intelligence, pp. 3633–3641, 2025. doi: 10.24963/ijcai.2025/404. URL https://www.ijcai.org/proceedings/2025/404.

Zhilin Zhao, Longbing Cao, and Yuanyu Wan. Mixture of online and offline experts for non-stationary time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 22867–22874, 2025. doi: 10.1609/aaai.v39i21.34448. URL https://ojs.aaai.org/index.php/AAAI/article/view/34448.

Guohui Zheng, Zongyu Yang, Songfen Liu, Rui Ma, Xinwen Gong, Ao Wang, Shuo Wang, and Wulyu Zhong. Real-time equilibrium reconstruction by multi-task learning neural network based on HL-3 tokamak. *Nuclear Fusion*, 64(12):126041, 2024. doi: 10.1088/1741-4326/ad8014.

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. AAAI Press, 2021. doi: 10.1609/aaai.v35i12.17325.

A SUPPLEMENTARY MATERIAL

This appendix provides additional experimental details and analyses supporting the results in the main paper. We describe the datasets, expert banks, routing implementations, and evaluation protocols, and then present supplementary analyses of forecasting errors, geometry metrics, routing behavior, and robustness. We further include an expanded related-work discussion and a summary of limitations and future directions.

A.1 DATASETS, PREPROCESSING, AND DATA SPLITS

MAST-U VDE simulation data for Section 4.2.1. We generate the MAST-U dataset using FreeGSNKE simulations of vertical-displacement events. The simulations cover different X-point and strike-point geometries, initial current conditions, and twelve-variable voltage inputs. Within each trajectory, the control voltages remain fixed, while they vary across trajectories. Each trajectory contains 21 state points from step 0 to step 20 and 20 control steps with $\Delta t = 0.5\,\mathrm{ms}$, resulting in a 10-ms recursive forecasting horizon. The state variables are organized into five physical heads. *Active* contains 12 active-coil or circuit currents, *Passive* contains 138 passive-structure currents, I_p denotes the scalar plasma current, ψ represents the poloidal magnetic-flux field on a 65×129 grid, and J_{tor} represents the corresponding toroidal-current-density field. We obtain 8,000 accepted trajectories in total. The trajectories are split by identity into 4,000 samples for expert training, 2,400 samples for router training, 800 samples for validation, and 800 samples for formal testing.

MAST experimental data for Section 4.2.2. We use experimental trajectories from the MAST device for the real-world evaluation. The data are obtained from the Level-2 all-shot archive, where each shot is stored as an individual Zarr record. We identify the flat-top interval from the plasma-current trace and extract continuous windows of at least 50 ms with finite and time-aligned measurements. Figure 5 illustrates this procedure for complete shot 29966 from the archive. The upper panel shows the I_p-based phase segmentation, with the central shaded interval marking the flat-top, while the lower panel shows the sampling grids of the diagnostic groups. This example makes the window extraction and time alignment explicit at the level of a single record.

The prediction state is divided into four physical heads. *Coil* contains ten poloidal-field-coil currents from P2IL, P2IU, P2OL, P2OU, P3L, P3U, P4L, P4U, P5L, and P5U. *Solenoid* contains the solenoid current, I_p denotes the plasma current, and ψ represents the poloidal-flux field on a 65×65 grid. The control input consists of four poloidal-field-coil voltages from P1, P2, P4, and P5 together with one NBI signal. The current, voltage, I_p, and NBI signals are sampled on a 0.25-ms grid, whereas ψ is sampled every 5 ms. The recursive forecast proceeds with a 5-ms step while preserving the original time axis of each shot. The strict shot-disjoint split contains 3,321 shots for expert training, 1,992 shots for router training, 664 shots for validation, and 664 shots for formal testing. We further define a common cohort of 367 shots with at least 20 valid recursive steps for the 100-ms endpoint evaluation.

A.2 TWO FROZEN EXPERT BANKS

MAST-U simulation expert bank. We use a fixed bank of three one-step experts, TF-NN, TFLinear-Fusion, and SINDyC-dual. Each expert receives the 151-dimensional physical state formed by 12 active-circuit currents, 138 passive-structure currents, and I_p, together with the twelve-variable voltage input and the two 65×129 fields ψ and J_{tor}. It predicts the next-step increments of five heads, including Active, Passive, I_p, ψ, and J_{tor}. TF-NN is trained as a direct data-driven predictor. TFLinearFusion keeps the TF-NN and linear proxy models frozen and learns gated fusion corrections, with a TF-NN fallback at the initial source step. SINDyC-dual keeps the TF-NN predictor frozen and learns a dual physics head-fusion correction. Expert fitting uses the 4,000 expert-training trajectories, while 800 validation trajectories are used for model selection. The 2,400 router-training trajectories and 800 formal-test trajectories remain outside expert fitting. All three experts are frozen before router training, and the router selects one expert independently for each physical head, giving $3^5 = 243$ joint expert combinations.

Real-MAST expert bank. The bank contains four frozen models, baseline, ODE0.25, ODE5, and Dual. The baseline receives 21 history samples of the 12-dimensional fast state, 40 control samples

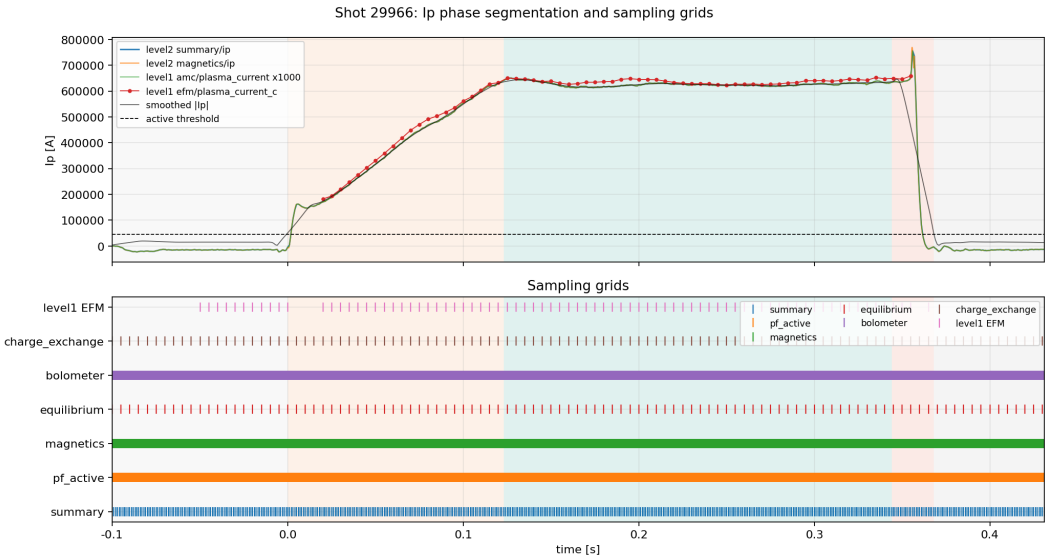

Figure 5: Illustrative complete MAST shot 29966 from the Level-2 archive. The upper panel shows the I_p-based phase segmentation, with the central shaded interval denoting the flat-top. The lower panel shows the sampling grids of the Level-1 and Level-2 diagnostic groups.

containing four coil voltages and NBI, and two 65×65 ψ frames. It predicts the next 20 fast-state samples and the next ψ frame. The expert bank separates the fast temporal signals from the slower two-dimensional flux field. ODE0.25 uses the 0.25-ms fast-state sequence to learn gated residual corrections for Coil, Solenoid, and I_p. ODE5 uses the 5-ms ψ dynamics together with a rank-eight ψ POD representation and the fast state to learn field and endpoint-state corrections. Dual enables both correction branches. Each derived expert freezes the baseline trunk and adds zero-initialized gated residual heads, so its initial output equals the baseline. Expert fitting uses 3,321 training shots and 664 validation shots, while 1,992 router-training shots and 664 formal-test shots remain outside expert fitting. The Coil, Solenoid, and I_p heads each have four candidate experts, while the ψ head has three because ODE0.25 does not produce a ψ correction. The resulting factorized action space is $4 \times 4 \times 4 \times 3 = 192$ joint combinations.

A.3 Router inputs, architectures, and training details

A.3.1 Router inputs

At forecast step t, the router receives an observation constructed from the predicted trajectory history, available control information, positional features, and expert-derived context:

$$o_t = g(\widehat{x}_{\leq t}, u_{\geq t}, p, \xi_t), \tag{16}$$

where $\widehat{x}_{\leq t}$ denotes the recursively predicted state history, $u_{\geq t}$ represents the available future control preview, p contains optional position and time-index features, and ξ_t summarizes information extracted from candidate experts. Based on this observation, the router selects experts for the next recursive transition, while the selected experts receive the current control input u_t. During training, teacher-forced observations use recorded state histories, whereas rollout-based training uses states predicted by the current router.

For MAST-U, the router observation is built from state representations, control previews, and expert predictions. The local observation includes a 128-dimensional baseline context, a 128-dimensional field context, a ten-dimensional summary of the baseline prediction, a 20-step preview of the twelve voltage controls, and a preview-validity mask. PPO-MoE and FIRE additionally use the full 20-step control sequence with its trajectory mask, the current position index, three position features, expert prediction summaries of size $3 \times 5 \times 4$, candidate validity masks, and five head-wise disagreement features. The routing heads correspond to Active, Passive, I_p, ψ, and J_{tor}. The flat routers operate

over $3^5 = 243$ joint expert combinations, whereas FIRE predicts a separate three-way categorical distribution for each head.

For real MAST, the observation is designed to capture both fast signal evolution and slower flux dynamics. The local observation contains a 128-dimensional baseline context, the predicted 20-step sequence of the 12 fast-state variables, a 128-dimensional embedding of the baseline ψ prediction, and up to ten control blocks with shape 20×5. Each control block contains twenty 0.25-ms samples from the four coil-voltage variables and NBI. The full-control branch further stores up to 128 recursive state steps with shape $128 \times 20 \times 5$, together with a trajectory mask, the current position index, three position features, four state-expert predictions of size $4 \times 20 \times 12$, three ψ expert embeddings of size 3×128, validity masks, and four head-wise disagreement features. The routing heads are Coil, Solenoid, I_p, and ψ. The flat routers use a joint action space of $(4, 4, 4, 3)$ with 192 combinations, while FIRE uses a factorized categorical policy over individual heads.

A.3.2 Router objectives and implementation details

We use the same state and action notation as in the main method. The state is partitioned into H physical heads,

$$x_t = (x_{t,1}, \ldots, x_{t,H}), \qquad x_{t,h} \in \mathbb{R}^{d_h}, \tag{17}$$

and head h has a frozen candidate set $\mathcal{E}_h$ with $K_h = |\mathcal{E}_h|$. The joint action space is $\mathcal{A} = \prod_{h=1}^{H} \mathcal{E}_h$. Once an action $a_t = (a_{t,1}, \ldots, a_{t,H})$ is selected, the frozen experts advance all heads recursively,

$$\widehat{x}_0 = x_0, \qquad \widehat{x}_{t+1,h} = f_{h,a_{t,h}}(\widehat{x}_t, u_t), \qquad t = 0, \ldots, T-1. \tag{18}$$

The routing policy is evaluated by the complete recursive objective

$$J(\pi) = \mathbb{E}_\tau \left[\mathcal{E}_{\mathrm{roll}} \left(\widehat{x}_{1:T}(\pi), x_{1:T} \right) \right], \tag{19}$$

where the expectation averages over trajectories or shots and $\mathcal{E}_{\mathrm{roll}}$ is the domain-specific complete-rollout error. Thus, an action changes both the next prediction and the states encountered by later routing decisions.

Baseline-MoE uses the local observation encoder and a flat categorical actor. For MAST-U, the baseline prediction summary and each control block are projected to 128 dimensions, the control sequence is encoded by a GRU, and the two context branches are fused by a two-layer 256-dimensional MLP with LayerNorm and SiLU activations. For real MAST, the 20-step fast-state prediction is encoded by a GRU, the control preview is encoded by a projection and GRU, and these features are fused with the baseline context through the same 256-dimensional representation. DaD-MoE keeps the corresponding encoder and actor and changes the state distribution through deterministic learner roll-ins. PPO-MoE adds a two-layer four-head Transformer with hidden dimension 128 for the complete control sequence, expert-aware state and field branches, and position metadata before the 256-dimensional fusion layer. Its actor remains flat with 243 MAST-U actions or 192 real-MAST actions. FIRE uses factorized head actors. The real-MAST `new-fire` model freezes the local encoder and joint behavior-cloned prior and initializes its new residual gate at zero. MAST-U FIRE first distills the joint prior into five head-wise policies and then updates the factorized policy during Stage A.

For head h and candidate expert e, the normalized one-step cost is

$$c_{h,e}(\widehat{x}_t, x_{t+1}) = \frac{1}{d_h} \sum_{j=1}^{d_h} \left(\frac{f_{h,e}(\widehat{x}_t, u_t)_j - x_{t+1,h,j}}{\sigma_{h,j}} \right)^2, \tag{20}$$

where $\sigma_{h,j}$ is estimated from the expert-training split and then kept fixed. The per-head oracle and additive joint cost are

$$a_h^* = \arg\min_{e \in \mathcal{E}_h} c_{h,e}, \qquad C(a) = \sum_{h=1}^{H} c_{h,a_h}. \tag{21}$$

Ties are resolved by selecting the smallest candidate index.

Baseline-MoE constructs teacher-forced observations and minimizes

$$\mathcal{L}_{\mathrm{BC}}(\theta) = -\mathbb{E}_{(o,a^*)} \log \pi_\theta(a^* \mid o), \tag{22}$$

where π_θ is a flat categorical policy over joint actions. DaD-MoE initializes from the Baseline-MoE router. At aggregation round k, the previous router generates

$$a_t^{(k)} = \arg\max_{a \in \mathcal{A}} \pi_{k-1}(a \mid o_t^{(k)}), \tag{23}$$

and the one-step oracle labels the resulting learner-induced observations. The round samples and aggregated dataset are

$$R_k = \left\{ \left(o_t^{(k)}, a_t^{*(k)}\right) \middle| \tau \in \mathcal{D}_{\text{router}}^{\text{train}},\ t = 0, \ldots, T_\tau - 1 \right\}, \qquad D_k = D_0 \cup \bigcup_{i=1}^{k} R_i. \tag{24}$$

PPO-MoE trains on complete recursive episodes. Its potential combines normalized prefix error and final-tail error

$$\Phi_t = 0.8\, \overline{\mathcal{E}}_{\text{all}}^{\text{train}}(t) + 0.2\, \overline{\mathcal{E}}_{\text{tail}}^{\text{train}}(t). \tag{25}$$

The tail term is zero before its evaluation window begins. Complete-episode Monte Carlo returns are computed with

$$r_t = -(\Phi_{t+1} - \Phi_t), \qquad G_t = \sum_{\ell=t}^{T-1} \gamma^{\ell - t} r_\ell, \qquad \gamma = 1. \tag{26}$$

The advantages are formed from $G_t - V_{\psi_{\text{old}}}(o_t)$ and normalized with the rollout weights

$$\widehat{A}_t = \text{Norm}_{\mathcal{B}}\left[G_t - V_{\psi_{\text{old}}}(o_t)\right], \qquad \rho_t(\theta) = \frac{\pi_\theta(a_t \mid o_t)}{\pi_{\text{old}}(a_t \mid o_t)}. \tag{27}$$

The clipped policy objective is

$$\mathcal{L}_{\text{clip}} = -\mathbb{E}_t \left[\min\left(\rho_t(\theta)\widehat{A}_t, \text{clip}\big(\rho_t(\theta), 1-\epsilon, 1+\epsilon\big)\widehat{A}_t \right) \right]. \tag{28}$$

The clipped value target is

$$V_t^{\text{clip}} = V_{\psi_{\text{old}}}(o_t) + \text{clip}\left(V_\psi(o_t) - V_{\psi_{\text{old}}}(o_t), -\epsilon_V, \epsilon_V\right), \tag{29}$$

and the value loss uses the larger of the unclipped and clipped SmoothL1 losses

$$\mathcal{L}_V = \frac{1}{2}\mathbb{E}_t \left[\max\left(\text{SmoothL1}\left(V_\psi(o_t), G_t\right), \text{SmoothL1}\left(V_t^{\text{clip}}, G_t\right) \right) \right]. \tag{30}$$

The normalized policy entropy is

$$\mathcal{H}(\pi_\theta) = \frac{\mathbb{E}_t\left[-\sum_a \pi_\theta(a \mid o_t) \log \pi_\theta(a \mid o_t)\right]}{\log |\mathcal{A}|}, \tag{31}$$

with the numerator replaced by the sum of factorized head entropies for FIRE. The complete PPO objective is

$$\mathcal{L}_{\text{PPO}} = \mathcal{L}_{\text{clip}} + \lambda_V \mathcal{L}_V - \lambda_H \mathcal{H}(\pi_\theta) + \alpha_{\text{aux}}(m)\mathcal{L}_{\text{aux}}, \tag{32}$$

where $\lambda_V = 0.5$ and λ_H is the adaptive entropy coefficient. The auxiliary term evaluates candidate potential increments, centers them across candidates, and applies the detached result under the current policy

$$\mathcal{L}_{\text{aux}} = \mathbb{E}_t \left[\sum_{a \in \mathcal{A}} \pi_\theta(a \mid o_t)\, \text{stopgrad}\left(\widetilde{\Delta\Phi}_t(a)\right) \right]. \tag{33}$$

In the implementation, the candidate costs are centered within each head and averaged over heads. This auxiliary is used only during the first 20% of PPO updates.

FIRE factorizes the policy across heads

$$\pi_\theta(a \mid o) = \prod_{h=1}^{H} \pi_{\theta,h}(a_h \mid o). \tag{34}$$

The factorized initialization is obtained from the exact marginals of the joint behavior-cloned policy

$$\pi_h^{\text{init}}(e \mid o) = \sum_{\substack{a \in \mathcal{A} \\ a_h = e}} \pi_{\text{BC}}(a \mid o). \tag{35}$$

At Stage A, FIRE computes the candidate regret

$$\delta_{h,e}(o) = c_{h,e}(\widehat{x}_t, x_{t+1}) - \min_{j \in \mathcal{E}_h} c_{h,j}(\widehat{x}_t, x_{t+1}) \tag{36}$$

and forms the prior-weighted mirror target

$$q_{k,h}(e \mid o) = \frac{\operatorname{stopgrad}\left[\pi_{k,h}(e \mid o)\right] \exp\left[-\delta_{h,e}(o)/s_h(o)\right]}{\sum_{j \in \mathcal{E}_h} \operatorname{stopgrad}\left[\pi_{k,h}(j \mid o)\right] \exp\left[-\delta_{h,j}(o)/s_h(o)\right]}. \tag{37}$$

The scale $s_h(o)$ is the median positive regret, with standard-deviation and unit-value fallbacks for degenerate rows. The Stage-A dataset $\mathcal{D}_{\mathrm{A}}^{(k)}$ contains the teacher-forced observations together with all learner-induced observations collected through round k. The parent policy is kept fixed while constructing each mirror target, and only the student policy is optimized with the following loss.

$$\mathcal{L}_{\mathrm{A}} = -\mathbb{E}_{o \sim \mathcal{D}_{\mathrm{A}}^{(k)}} \left[\frac{1}{H} \sum_{h=1}^{H} \sum_{e \in \mathcal{E}_h} q_{k,h}(e \mid o) \log \pi_{\theta,h}(e \mid o) \right]. \tag{38}$$

Let π_{A} denote the selected Stage-A policy. Stage B disables the immediate-cost auxiliary and adds the normalized per-head reference divergence. The Stage-B dataset $\mathcal{D}_{\mathrm{B}}$ contains the observations from complete recursive rollouts predicted by the selected Stage-A policy and its Stage-B descendants. The Stage-A policy is treated as the reference distribution during Stage-B optimization.

$$\mathcal{L}_{\mathrm{anchor}} = \mathbb{E}_{o \sim \mathcal{D}_{\mathrm{B}}} \left[\frac{\sum_{h=1}^{H} D_{\mathrm{KL}}\left(\pi_{\theta,h}(\cdot \mid o) \| \pi_{\mathrm{A},h}(\cdot \mid o)\right)}{\sum_{h=1}^{H} \log |\mathcal{E}_h|} \right]. \tag{39}$$

The Stage-B objective is

$$\mathcal{L}_{\mathrm{B}} = \mathcal{L}_{\mathrm{clip}} + \lambda_V \mathcal{L}_V - \lambda_H \mathcal{H}(\pi_\theta) + \beta \mathcal{L}_{\mathrm{anchor}}. \tag{40}$$

The denominator in the reference term equals $\log |\mathcal{A}|$ for the factorized action space.

A.3.3 Training and checkpoint selection

All router features are normalized using statistics from the expert-training split. The expert parameters remain fixed during router training and evaluation. MAST-U experiments use three random seeds (20260813, 20260814, and 20260815), while real-MAST experiments use three seeds (20260724, 20260725, and 20260726). The formal-test split is reserved for final evaluation.

Baseline-MoE and each DaD-MoE aggregation round use AdamW with learning rate 3×10^{-4}, weight decay 10^{-4}, batch size 512, mixed precision, gradient clipping at 0.5, and a maximum of 200 supervised epochs. Early stopping uses patience 20 with a minimum validation improvement of 10^{-5}. DaD-MoE uses ten deterministic argmax roll-in rounds, resets AdamW at each round, aggregates new learner-state rows with all earlier rows, and warm-starts the router from the preceding checkpoint. Baseline-MoE selects its checkpoint by teacher-forced validation cross-entropy. DaD-MoE selects the best checkpoint within each round by teacher-forced validation cross-entropy and then selects among completed rounds by complete-rollout validation error.

PPO-MoE starts from the paired Baseline-MoE checkpoint and keeps the expert bank unchanged. It collects complete physical episodes with 48 parallel environments until at least 8,192 transitions are available. The actor and critic learning rates are 2×10^{-5} and 3×10^{-4}, respectively. Each update uses three PPO epochs, minibatches of 512, weight decay 10^{-4}, gradient clipping at 0.5, $\epsilon = \epsilon_V = 0.2$, $\lambda_V = 0.5$, $\gamma = 1$, and a target-KL threshold of 0.01. The critic is pretrained for five epochs. The immediate-cost auxiliary starts at 0.1 and is removed linearly during the first 20% of updates. The entropy coefficient is initialized at 0.002 and adapted toward the prescribed entropy target. Validation is run every ten updates with patience ten, and the checkpoint with the lowest complete-rollout validation error is retained.

FIRE first matches the frozen behavior-cloned policy through exact analytic marginals for real MAST. For MAST-U, the marginal policy is obtained by a 50-epoch distillation stage with batch size 512, learning rate 3×10^{-4}, and early stopping patience 5. Stage A uses ten to twenty soft-mirror rounds with 200 epochs per round, batch size 512, AdamW learning rate 3×10^{-4}, weight decay 10^{-4},

mixed precision, gradient clipping at 0.5, and patience 20. After the minimum ten rounds, outer validation can stop Stage A after three consecutive rounds without improvement, with twenty rounds as the maximum. Learner rollouts use deterministic per-head argmax actions, and completed rounds are compared by complete-rollout validation error.

Stage B follows the PPO optimization procedure with complete recursive episodes. It uses 48 parallel environments, 8,192 transitions per update, three PPO epochs, minibatches of 512, actor and critic learning rates of 2×10^{-5} and 3×10^{-4}, $\epsilon = \epsilon_V = 0.2$, $\lambda_V = 0.5$, $\gamma = 1$, and target KL threshold 0.02. The critic is pretrained for five epochs, the immediate-cost auxiliary is disabled, and adaptive entropy regularization is retained. The anchoring coefficient is selected from $\{0.01, 0.05, 0.1\}$ using validation data. The reported FIRE configurations use $\beta = 0.05$ for real MAST and $\beta = 0.1$ for MAST-U.

At inference time, flat routers select the highest-probability joint action, whereas FIRE selects the highest-probability expert independently for each head. All reported results use checkpoints selected from validation procedures.

A.4 Supplementary results for the two main experiments

This section provides additional quantitative results for the two experiments introduced in the main paper. The reported metrics follow the evaluation protocols used in each experiment. MAST results are measured using pooled NRMSE, whereas MAST-U physical results use the equal-weight Macro relative-ℓ_2 error. Geometry results are reported separately for each observable because these variables have different physical units. In all tables, positive percentages denote error reductions relative to Baseline-MoE.

A.4.1 Real-MAST error tables

Tables 3 and 4 provide supplementary comparisons of real-history one-step prediction and the 100-ms endpoint rollout. The one-step errors are similar across routing methods. FIRE-MoE with $\beta = 0.05$ achieves a Macro error of 0.01263718, compared with 0.01266768 for Baseline-MoE, corresponding to a 0.24% reduction. On the 100-ms endpoint evaluation using the common 367-shot cohort, FIRE-MoE reduces the Macro error from 0.08250709 to 0.07762687, corresponding to a 5.91% reduction. The largest head-wise improvements are observed for I_p and ψ. These results are consistent with the observation in the main paper that one-step accuracy does not necessarily determine long-horizon rollout performance. They also show that the proposed routing frameworks improve recursive forecasting compared with the behavior-cloning baseline.

Table 3: Real-MAST one-step physical error

Model	Coil rel.%	Solenoid rel.%	I_p rel.%	ψ rel.%	Macro rel.%
Baseline expert	0.01145627 -4.55%	0.00764136 -26.31%	0.01164637 -8.97%	0.02347162 -2.16%	0.01355390 -7.00%
ODE0.25 expert	0.01057540 +3.48%	0.00654001 -8.11%	0.01072519 -0.35%	0.02347162 -2.16%	0.01282806 -1.27%
ODE5 expert	0.01137604 -3.82%	0.00747783 -23.61%	0.01162176 -8.74%	<u>0.02267818</u> +1.30%	0.01328845 -4.90%
Dual expert	<u>0.01054284</u> +3.78%	<u>0.00652607</u> -7.88%	<u>0.01071655</u> -0.27%	0.02283858 +0.60%	<u>0.01265601</u> +0.09%
Baseline-MoE	0.01095724 +0.00%	**0.00604951** +0.00%	**0.01068774** +0.00%	0.02297624 +0.00%	0.01266768 +0.00%
E3-6 / PPO-MoE	0.01100650 -0.45%	0.00631579 -4.40%	0.01091452 -2.12%	0.02293685 +0.17%	0.01279341 -0.99%
DAD-MoE	0.01092384 +0.30%	0.00608704 -0.62%	0.01073178 -0.41%	0.02289056 +0.37%	0.01265830 +0.07%
FIRE-MoE ($\beta = 0.05$)	**0.01079566** +1.47%	0.00611954 -1.16%	0.01074938 -0.58%	**0.02288415** +0.40%	**0.01263718** +0.24%
FIRE-MoE ($\beta = 0.01$)	0.01079887 +1.45%	0.00611329 -1.05%	0.01074562 -0.54%	0.02289171 +0.37%	0.01263737 +0.24%

Table 4: Real-MAST 100 ms endpoint physical error

Model	Coil rel.%	Solenoid rel.%	I_p rel.%	ψ rel.%	Macro rel.%
Baseline expert	0.13641323 -16.26%	0.14592502 -104.69%	0.07965854 -20.40%	0.08479698 -12.70%	0.11169844 -35.38%
ODE0.25 expert	0.11983328 -2.13%	0.08113502 -13.81%	0.06786844 -2.58%	0.08346238 -10.93%	0.08807478 -6.75%
ODE5 expert	0.11917460 -1.57%	<u>0.07387084</u> -3.62%	<u>0.06592795</u> +0.36%	<u>0.07940585</u> -5.54%	<u>0.08459481</u> -2.53%
Dual expert	<u>0.11557822</u> +1.50%	0.08661927 -21.50%	0.06781532 -2.50%	0.07989563 -6.19%	0.08747711 -6.02%
Baseline-MoE	0.11733437 +0.00%	0.07129198 +0.00%	0.06616335 +0.00%	0.07523867 +0.00%	0.08250709 +0.00%
E3-6 / PPO-MoE	0.11826300 -0.79%	0.06751042 +5.30%	0.06383840 +3.51%	0.07523844 +0.00%	0.08121256 +1.57%
DAD-MoE	0.11355189 +3.22%	**0.06586666** +7.61%	0.06056208 +8.47%	0.07122757 +5.33%	0.07780205 +5.70%
FIRE-MoE ($\beta = 0.05$)	0.11331428 +3.43%	0.06832598 +4.16%	**0.05891582** +10.95%	**0.06995139** +7.03%	0.07762687 +5.91%
FIRE-MoE ($\beta = 0.01$)	**0.11157462** +4.91%	0.06845868 +3.97%	0.05963361 +9.87%	0.07074357 +5.97%	**0.07760262** +5.94%

Figure 6 provides a representative long-horizon example from the formal MAST test set. The left block compares the measured and FIRE-predicted values for the ten coil currents, the solenoid current, and the plasma current on the recursive 5-ms grid. The right block shows the measured poloidal-flux field, the FIRE prediction, and the signed residual at 100 ms. This example makes the agreement between channel-wise tracking and field-level prediction visible for a single complete shot.

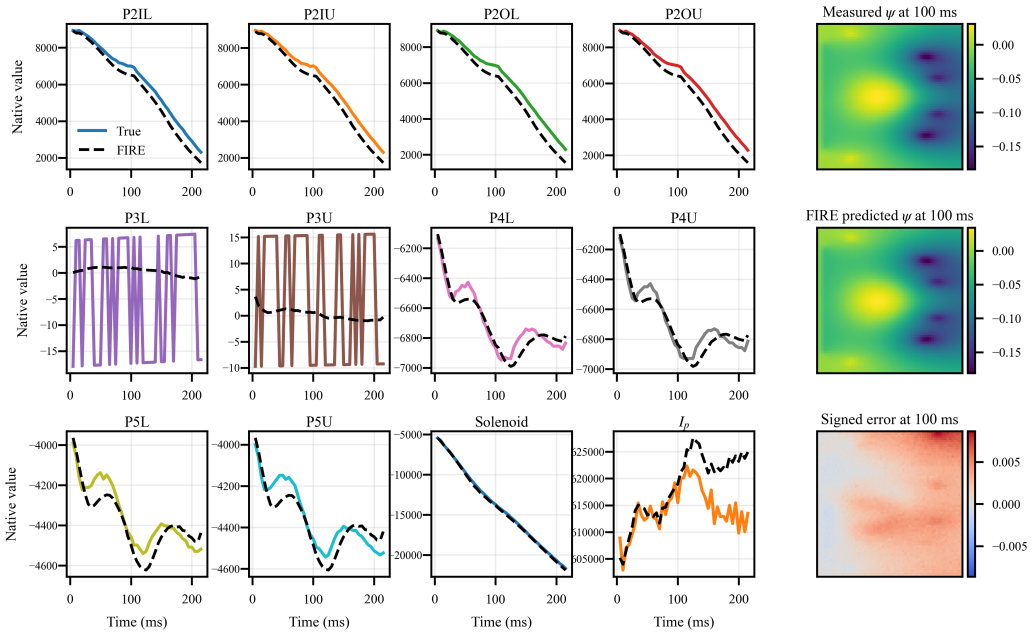

Figure 6: Representative FIRE recursive forecast for MAST shot 16618. The left block compares measured and predicted fast-state channels on the recursive 5-ms grid. The right block shows the measured and predicted ψ fields and their signed residual at 100 ms.

A.4.2 MAST-U PHYSICAL ERROR TABLES

Tables 5 and 6 provide supplementary physical-head results for the MAST-U simulation. At one step, the routing methods remain close to Baseline-MoE. Baseline-MoE achieves a Macro error of 2.9668, while FIRE-MoE with $\beta = 0.1$ obtains a similar error of 2.9769. In contrast, the difference

Table 5: MAST-U one-step physical error

Model	Active rel.%	Passive rel.%	I_p rel.%	ψ rel.%	J_{tor} rel.%	Macro rel.%
TFLinearFusion	3.7882 -1.64%	1.4474 -1.40%	<u>3.5902</u> -2.13%	1.1519 -12.11%	<u>3.8602</u> -0.43%	3.0098 -1.45%
SINDyC-dual	<u>3.7689</u> -1.12%	<u>1.4397</u> -0.86%	3.5955 -2.28%	<u>1.0330</u> -0.54%	4.4494 -15.76%	3.0037 -1.24%
TF-NN	4.1376 -11.02%	1.5076 -5.61%	3.6975 -5.18%	1.2399 -20.67%	5.0418 -31.17%	3.1564 -6.39%
Baseline-MoE	**3.7271 0.00%**	**1.4274 0.00%**	3.5154 0.00%	**1.0275 0.00%**	3.8437 0.00%	2.9668 0.00%
E3-6-MOE	3.7764 -1.32%	1.4329 -0.38%	3.5361 -0.59%	1.0365 -0.88%	**3.8319 +0.31%**	2.9777 -0.37%
DAD-MOE	3.8289 -2.73%	1.4400 -0.88%	3.4986 +0.48%	1.0508 -2.27%	4.0403 -5.11%	2.9964 -1.00%
FIRE-MoE ($\beta = 0.1$)	3.7350 -0.21%	1.4296 -0.15%	**3.4646 +1.45%**	1.0407 -1.28%	4.1240 -7.29%	2.9769 -0.34%

Table 6: MAST-U full recursive physical error

Model	Active rel.%	Passive rel.%	I_p rel.%	ψ rel.%	J_{tor} rel.%	Macro rel.%
TFLinearFusion	<u>1.1937</u> +4.93%	<u>1.7057</u> +3.43%	<u>1.0028</u> +3.98%	<u>2.3470</u> +3.42%	<u>5.7921</u> +4.59%	3.6182 +3.48%
SINDyC-dual	1.2637 -0.65%	1.7657 +0.03%	1.0665 -2.13%	2.4488 -0.77%	6.3414 -4.46%	3.7538 -0.14%
TF-NN	1.2034 +4.16%	1.7630 +0.19%	1.0449 -0.06%	2.4859 -2.29%	6.5995 -8.71%	3.7526 -0.11%
Baseline-MoE	1.2556 0.00%	1.7663 0.00%	1.0443 0.00%	2.4301 0.00%	6.0707 0.00%	3.7486 0.00%
E3-6-MOE	**1.1370 +9.45%**	**1.6546 +6.32%**	**0.9788 +6.27%**	**2.2233 +8.51%**	**5.6783 +6.46%**	3.5095 +6.38%
DAD-MOE	1.1738 +6.51%	1.6941 +4.09%	0.9912 +5.08%	2.3084 +5.01%	5.8273 +4.01%	3.5941 +4.12%
FIRE-MoE ($\beta = 0.1$)	1.1736 +6.53%	1.6902 +4.31%	0.9935 +4.86%	2.2622 +6.91%	5.7353 +5.52%	3.5837 +4.40%

becomes larger during the complete 0.5–10-ms recursive rollout. PPO-MoE reduces the Macro error from 3.7486 to 3.5095, while FIRE-MoE reduces it to 3.5837, corresponding to reductions of 6.38% and 4.40%, respectively. These results are consistent with the observation in the main paper that local prediction accuracy does not necessarily determine long-horizon rollout performance. They further indicate that adaptive routing can exploit complementary expert behaviors during recursive forecasting. The 10-ms endpoint physical comparison is reported in the main text.

A.4.3 MAST-U GEOMETRY ERROR TABLES

Definitions of the geometry observables. We evaluate the predicted magnetic geometry using seven observables that characterize X-point localization, vertical motion, and magnetic topology. Let $(R_{X,l}, Z_{X,l})$ and $(R_{X,u}, Z_{X,u})$ denote the lower and upper X-point coordinates, and let (R_O, Z_O) denote the primary O-point location. The localization errors of the two X-points are defined as

$$d_{X,l} = \sqrt{(\widehat{R}_{X,l} - R_{X,l})^2 + (\widehat{Z}_{X,l} - Z_{X,l})^2}, \qquad d_{X,u} = \sqrt{(\widehat{R}_{X,u} - R_{X,u})^2 + (\widehat{Z}_{X,u} - Z_{X,u})^2}. \tag{41}$$

The vertical current-centroid position is computed from the toroidal current-density field as

$$Z_J = \frac{\int Z J_{\mathrm{tor}}(R,Z)\,\mathrm{d}R\,\mathrm{d}Z}{\int J_{\mathrm{tor}}(R,Z)\,\mathrm{d}R\,\mathrm{d}Z}, \qquad v_Z(t) = \frac{Z_J(t+\Delta t) - Z_J(t)}{\Delta t}, \tag{42}$$

where v_Z is evaluated with the native $\Delta t = 0.5\,\mathrm{ms}$ step. The O-point localization error is

$$d_O = \sqrt{(\widehat{R}_O - R_O)^2 + (\widehat{Z}_O - Z_O)^2}, \qquad S_X = Z_{X,u} - Z_{X,l}, \qquad d_{X,\mathrm{pair}} = \sqrt{\frac{d_{X,l}^2 + d_{X,u}^2}{2}}. \tag{43}$$

Here, S_X represents the vertical separation between the two X-points, while $d_{X,\mathrm{pair}}$ summarizes their joint localization error. The same critical-point procedure is applied to both ground-truth and predicted poloidal-flux fields, and the upper and lower X-point branches are matched according to their vertical positions. Position errors are reported in centimeters, Z_J and S_X are reported in centimeters, and v_Z is reported in meters per second.

The scalar localization errors measure the distance between predicted and reference positions but do not retain the direction of the displacement. Figure 7 compares the signed radial and vertical residuals of FIRE and the other methods at the first 0.5-ms prediction step on the same 800 paired MAST-U test trajectories. The three panels show the O-point, upper X-point, and lower X-point. Solid and dashed contours enclose 50% and 90% of the empirical residual distribution, respectively, and crosses mark the mean signed bias.

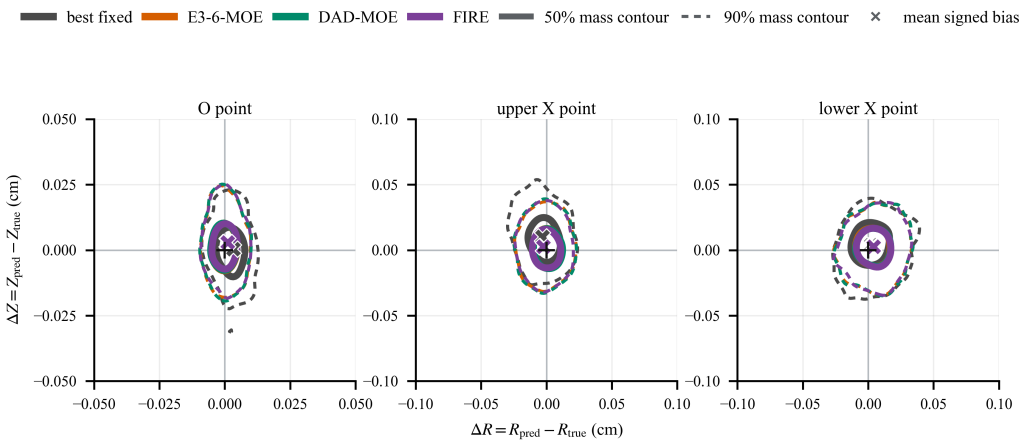

Figure 7: Signed O-point and X-point localization residuals at the first 0.5-ms prediction step over 800 paired MAST-U test trajectories. FIRE is shown together with the fixed expert, E3-6-MOE, and DAD-MOE using the same residual coordinates and contour definitions.

Tables 7, 8, and 9 report the seven geometry observables at one step, over the complete 0.5–10-ms rollout, and at the 10-ms endpoint. Because these observables have different physical units, each variable is reported separately rather than combined into a cross-variable Macro score. The one-step errors are close across routing methods. During the recursive rollout, however, FIRE-MoE achieves consistent improvements across all seven geometry observables. It reduces $d_{X,l}$, $d_{X,u}$, Z_J, v_Z, d_O, S_X, and $d_{X,\mathrm{pair}}$ by 7.73%, 7.02%, 8.90%, 6.27%, 9.65%, 14.79%, and 7.54%, respectively. At the 10-ms endpoint, the corresponding reductions are 9.88%, 7.51%, 10.77%, 10.15%, 11.38%, 18.54%, and 9.25%. These results show that the routing policy improves not only aggregate prediction errors but also physically meaningful geometric variables during long-horizon evolution.

Table 7: MAST-U one-step geometry error

Model	$d_{X,l}$ / cm rel.%	$d_{X,u}$ / cm rel.%	Z_J / cm rel.%	v_Z / m/s rel.%	d_O / cm rel.%	S_X / cm rel.%	$d_{X,\mathrm{pair}}$ / cm rel.%
TFLinearFusion	0.1998 +0.62%	0.2363 +0.58%	0.3372 -0.01%	6.7443 -0.01%	0.2998 +0.33%	0.0980 +4.35%	0.2188 +0.60%
SINDyC-dual	0.2013 -0.10%	0.2378 -0.06%	0.3314 +1.71%	6.6287 +1.71%	0.3008 -0.01%	0.1029 -0.38%	0.2203 -0.08%
TF-NN	0.2024 -0.69%	0.2365 +0.49%	0.3315 +1.68%	6.6302 +1.68%	0.2982 +0.87%	0.1022 +0.23%	0.2202 -0.00%
Baseline-MoE	0.2010 0.00%	0.2377 0.00%	0.3372 0.00%	6.7438 0.00%	0.3008 0.00%	0.1025 0.00%	0.2201 0.00%
E3-6-MOE	0.2012 -0.09%	0.2378 -0.05%	0.3372 0.00%	6.7438 0.00%	0.3008 -0.01%	0.1028 -0.31%	0.2203 -0.07%
DAD-MOE	0.2020 -0.45%	0.2360 +0.72%	0.3314 +1.73%	6.6274 +1.73%	0.2981 +0.90%	0.1034 -0.92%	0.2196 +0.23%
FIRE-MoE ($\beta = 0.1$)	0.2018 -0.38%	0.2359 +0.77%	0.3314 +1.73%	6.6272 +1.73%	0.2981 +0.88%	0.1004 +2.05%	0.2195 +0.29%

Table 8: MAST-U full recursive geometry error

Model	$d_{X,l}$ / cm rel.%	$d_{X,u}$ / cm rel.%	Z_J / cm rel.%	v_Z / m/s rel.%	d_O / cm rel.%	S_X / cm rel.%	$d_{X,\mathrm{pair}}$ / cm rel.%
TFLinearFusion	1.1495 +5.73%	0.6896 +6.74%	1.3104 +7.76%	7.8408 +5.32%	1.2102 +8.73%	0.9549 +10.67%	0.9478 +6.00%
SINDyC-dual	1.2324 -1.07%	0.7547 -2.06%	1.4362 -1.09%	8.0955 +2.25%	1.3400 -1.07%	1.0666 +0.22%	1.0219 -1.34%
TF-NN	1.1741 +3.71%	0.7124 +3.66%	1.3309 +6.32%	7.7769 +6.10%	1.2276 +7.41%	0.9734 +8.94%	0.9711 +3.70%
Baseline-MoE	1.2194 0.00%	0.7394 0.00%	1.4207 0.00%	8.2817 0.00%	1.3259 0.00%	1.0689 0.00%	1.0084 0.00%
E3-6-MOE	1.1277 +7.52%	0.6853 +7.32%	1.2947 +8.87%	7.7172 +6.82%	1.1994 +9.54%	0.9248 +13.49%	0.9331 +7.47%
DAD-MOE	1.1380 +6.68%	0.6917 +6.45%	1.3101 +7.78%	7.7356 +6.59%	1.2190 +8.06%	0.9280 +13.18%	0.9417 +6.62%
FIRE-MoE ($\beta = 0.1$)	1.1251 +7.73%	0.6875 +7.02%	1.2943 +8.90%	7.7625 +6.27%	1.1980 +9.65%	0.9109 +14.79%	0.9323 +7.54%

Table 9: MAST-U 10 ms endpoint geometry error

Model	$d_{X,l}$ / cm rel.%	$d_{X,u}$ / cm rel.%	Z_J / cm rel.%	v_Z / m/s rel.%	d_O / cm rel.%	S_X / cm rel.%	$d_{X,\mathrm{pair}}$ / cm rel.%
TFLinearFusion	2.7915 +7.08%	1.6408 +7.97%	3.3519 +8.97%	13.6660 +6.39%	3.1215 +9.82%	2.4966 +15.55%	2.2896 +7.31%
SINDyC-dual	3.0007 +0.11%	1.7967 -0.78%	3.6862 -0.11%	14.1724 +2.92%	3.4637 -0.06%	2.9200 +1.22%	2.4731 -0.12%
TF-NN	2.8185 +6.18%	1.6728 +6.17%	3.3763 +8.30%	13.3359 +8.65%	3.1427 +9.21%	2.5119 +15.03%	2.3176 +6.18%
Baseline-MoE	3.0041 0.00%	1.7828 0.00%	3.6820 0.00%	14.5983 0.00%	3.4616 0.00%	2.9562 0.00%	2.4701 0.00%
E3-6-MOE	2.7317 +9.07%	1.6173 +9.28%	3.2918 +10.60%	13.1415 +9.98%	3.0704 +11.30%	2.4285 +17.85%	2.2447 +9.12%
DAD-MOE	2.7603 +8.11%	1.6341 +8.34%	3.3209 +9.81%	13.1858 +9.68%	3.1057 +10.28%	2.4353 +17.62%	2.2682 +8.17%
FIRE-MoE ($\beta = 0.1$)	2.7073 +9.88%	1.6490 +7.51%	3.2853 +10.77%	13.1160 +10.15%	3.0677 +11.38%	2.4080 +18.54%	2.2415 +9.25%

A.5 FIRE Mechanism Analysis and Reference-Weight Robustness

FIRE combines two complementary components: full-information supervision over candidate experts and rollout refinement with a reference policy constraint. We analyze how these components contribute to recursive forecasting through matched ablation studies. The ablations focus on three questions: whether rollout refinement provides additional gains beyond Stage A supervision, whether the reference constraint affects refinement stability, and whether retaining relative expert quality through soft targets improves routing compared with hard expert labels.

We first evaluate these design choices on MAST-U, where the longer recursive rollout makes differences between routing strategies more apparent. All variants follow the same expert bank, router architecture, data split, and evaluation protocol as the main MAST-U experiment. Stage A uses learner-state observations with full-information soft targets derived from candidate expert costs. Full FIRE further performs Stage-B rollout refinement using the selected reference coefficient. The No-KL variant removes the reference constraint during Stage B, while the soft-versus-hard comparison replaces the soft target with a one-hot expert label.

Table 10 summarizes the matched comparisons. Full FIRE reduces the complete-rollout Macro error from 0.037383 ± 0.000582 for Stage A to 0.036495 ± 0.000337, corresponding to a 2.375% reduction. Removing the reference constraint results in a smaller improvement of 0.790%, suggesting that the reference term contributes to the refinement process in this setting. The soft-target comparison further shows a 4.413% improvement over hard labels, indicating that preserving relative candidate quality provides useful information beyond selecting only the lowest-cost expert.

We further examine the reference-weight choice in Stage B on real MAST. Rather than treating the anchoring coefficient as a fixed implementation detail, we evaluate whether the refinement behavior is sensitive to the strength of the constraint. The main experiment uses $\beta = 0.05$, and we compare it with a weaker constraint of $\beta = 0.01$ while keeping the expert bank, router architecture, data splits, normalization, and evaluation protocol unchanged.

Tables 3 and 4 show that the two coefficients produce nearly identical one-step and endpoint performance. With $\beta = 0.01$, FIRE achieves a Macro NRMSE of 0.01263737 at one step and 0.07760262 at the 100-ms endpoint, compared with 0.01263718 and 0.07762687 for $\beta = 0.05$. These results correspond to similar reductions relative to Baseline-MoE, suggesting that FIRE is not sensitive to small changes in the anchoring strength.

The complete recursive rollout results in Table 11 show the same trend. FIRE with $\beta = 0.05$ reduces the Macro NRMSE from 0.07057147 to 0.06506154, while $\beta = 0.01$ achieves 0.06530286. Both settings provide comparable improvements over Baseline-MoE. We therefore retain $\beta = 0.05$ as the main configuration and use $\beta = 0.01$ as a robustness check.

Table 10: FIRE mechanism ablations on complete recursive rollouts. The Macro NRMSE column reports the first and second methods in each comparison. Positive relative gains favor the first method. MAST-U and MAST use three-seed mean and population standard deviation. Soft-versus-hard rows are matched single-seed checks.

Domain	Comparison	Macro NRMSE	Relative gain (%)	Paired 95% CI
MAST-U	Full vs Stage A	0.036495 ± 0.000337 / 0.037383 ± 0.000582	$+2.375$	$[0.000132, 0.001898]$
MAST-U	Full vs No-KL	0.036495 ± 0.000337 / 0.036786 ± 0.000362	$+0.790$	$[-0.000210, 0.000899]$
MAST-U	Soft vs Hard	matched check	$+4.413$	–
MAST	Full vs Stage A	0.06649 ± 0.00038 / 0.06686 ± 0.00012	$+0.558$	$[-0.000109, 0.000864]$
MAST	Full vs No-KL	0.06649 ± 0.00038 / 0.06650 ± 0.00062	$+0.012$	$[-0.000365, 0.000418]$
MAST	Soft vs Hard	matched check	-1.963	–

Table 11: Probability-ensemble NRMSE over the complete recursive rollout on real MAST. Percentages in parentheses denote reductions relative to Baseline-MoE.

Model	Coil	Solenoid	I_p	ψ	Macro
Baseline-MoE	0.08931343	0.06128494	0.05566874	0.07601877	0.07057147
FIRE ($\beta = 0.05$)	0.08422290 (5.70%)	0.05676223 (7.38%)	0.04713915 (15.32%)	0.07212188 (5.13%)	0.06506154 (7.81%)
FIRE ($\beta = 0.01$)	0.08390396 (6.06%)	0.05738281 (6.37%)	0.04746427 (14.74%)	0.07246041 (4.68%)	0.06530286 (7.47%)

A.6 Analyzing Expert Specialization through Routing Entropy

We analyze routing probabilities and normalized entropy to understand how the routers exploit expert specialization during recursive forecasting. For each physical head h, the normalized entropy is computed as $H(p_h)/\ln K_h$ according to Eq. 31, where lower values indicate more concentrated expert selection. For flat routers, the joint action probabilities are first marginalized to obtain per-head routing distributions, whereas FIRE directly uses its factorized head-wise probabilities. All curves are computed on validation trajectories using the same horizon-wise aggregation as the corresponding routing diagnostics.

MAST-U simulation. Figure 8 shows the routing probabilities and normalized entropy for the four MAST-U routers over the complete recursive rollout. The fourth row corresponds to FIRE with $\beta = 0.1$. The MAST-U expert bank contains TFLinearFusion, SINDyC-dual, and TF-NN.

The strongest routing differences appear in ψ and J_{tor}. For ψ, Baseline-MoE and PPO-MoE assign approximately 0.90 probability to SINDyC-dual, with normalized entropies of 0.259 and 0.238. DaD-MoE distributes probability more broadly across the three experts, reducing the SINDyC-dual probability to 0.699 and increasing entropy to 0.582. FIRE retains SINDyC-dual as the dominant expert while allowing more balanced selection, with a mean probability of 0.747 and entropy of 0.417.

A similar pattern appears for J_{tor}. Baseline-MoE and PPO-MoE strongly favor TFLinearFusion, with probabilities near 0.92 and entropies of 0.102 and 0.135. In contrast, DaD-MoE and FIRE distribute probability between TFLinearFusion and SINDyC-dual, with entropies of 0.589 and 0.501. These different routing behaviors show that the methods exploit the available expert diversity in different ways, even though all routers improve over Baseline-MoE in the main MAST-U error comparison.

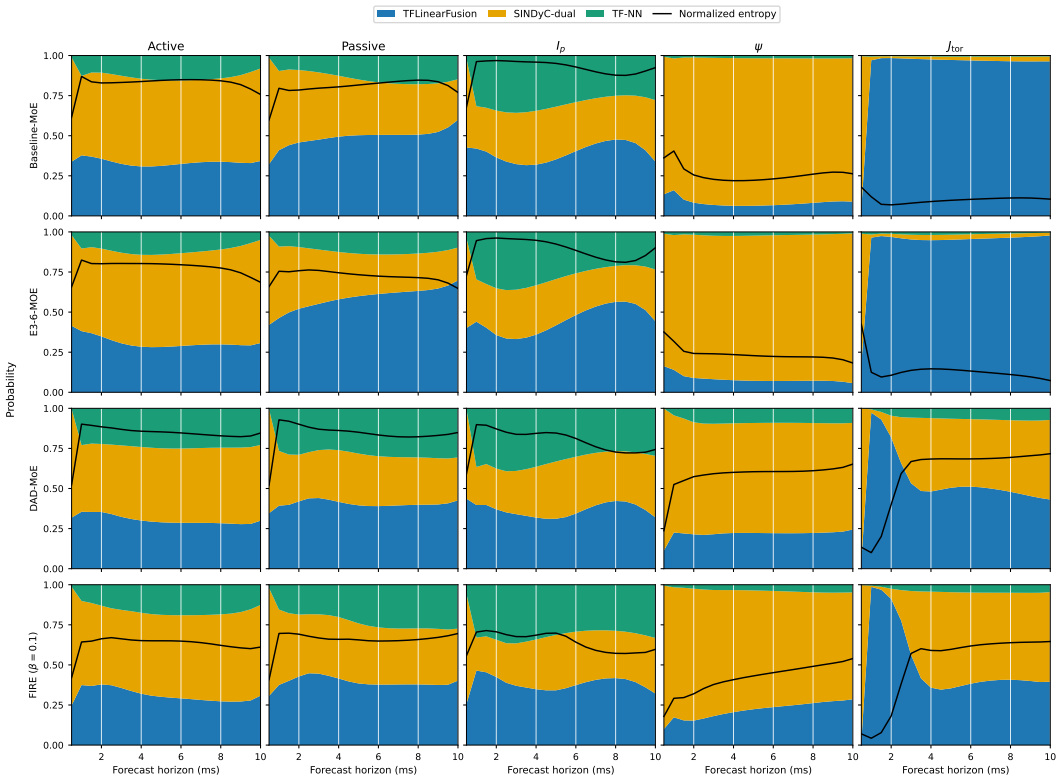

Figure 8: Validation-only routing probabilities and normalized per-head entropy for the four MAST-U routers over the complete 10-ms recursive rollout. The rows show Baseline-MoE, E3-6-MOE, DaD-MoE, and FIRE with $\beta = 0.1$. The columns correspond to Active, Passive, I_p, ψ, and J_{tor}.

The Active and Passive heads exhibit smaller routing changes. FIRE assigns approximately 0.53 probability to SINDyC-dual for Active and distributes Passive probability mainly between TFLinear-Fusion and SINDyC-dual, resulting in entropies of 0.629 and 0.655. The I_p routing distributions are also relatively similar across methods, although FIRE reduces entropy from 0.918 for Baseline-MoE to 0.640. These results suggest that routing specialization is most pronounced when the candidate experts provide distinct predictive behaviors for a specific physical variable.

Real MAST. Figure 9 shows the routing probabilities and normalized entropy for the four MAST routers over the complete recursive rollout. The PPO-MoE trace is labeled E3-6-MOE in the source plot, and the fourth row corresponds to FIRE with $\beta = 0.05$.

For the Coil head, the three flat routers produce similar routing distributions. Their mean expert probabilities are approximately $(0.247, 0.278, 0.219, 0.256)$ for Baseline-MoE, $(0.231, 0.250, 0.261, 0.259)$ for PPO-MoE, and $(0.232, 0.268, 0.236, 0.264)$ for DaD-MoE, with normalized entropies of 0.992, 0.989, and 0.962. FIRE shifts more probability toward Dual, resulting in mean probabilities of $(0.220, 0.257, 0.229, 0.294)$ and entropy of 0.871. Although the flat routers exhibit similar allocations, FIRE achieves the largest Coil improvement in Table 2. This indicates that routing concentration alone does not determine forecasting quality. Instead, the benefit comes from assigning higher probability to experts that better match the dynamics of a specific physical variable.

The largest differences among MAST routers appear in Solenoid. DaD-MoE increases the ODE0.25 probability to approximately 0.490 and decreases the ODE5 probability to 0.114, compared with 0.347 and 0.165 for Baseline-MoE. Its entropy decreases to 0.649 from 0.833 for Baseline-MoE and 0.823 for PPO-MoE. FIRE also forms a more concentrated distribution with probabilities $(0.226, 0.354, 0.162, 0.258)$ and entropy 0.674. The stronger specialization in Solenoid corresponds to the larger improvements achieved by DaD-MoE and FIRE for this head.

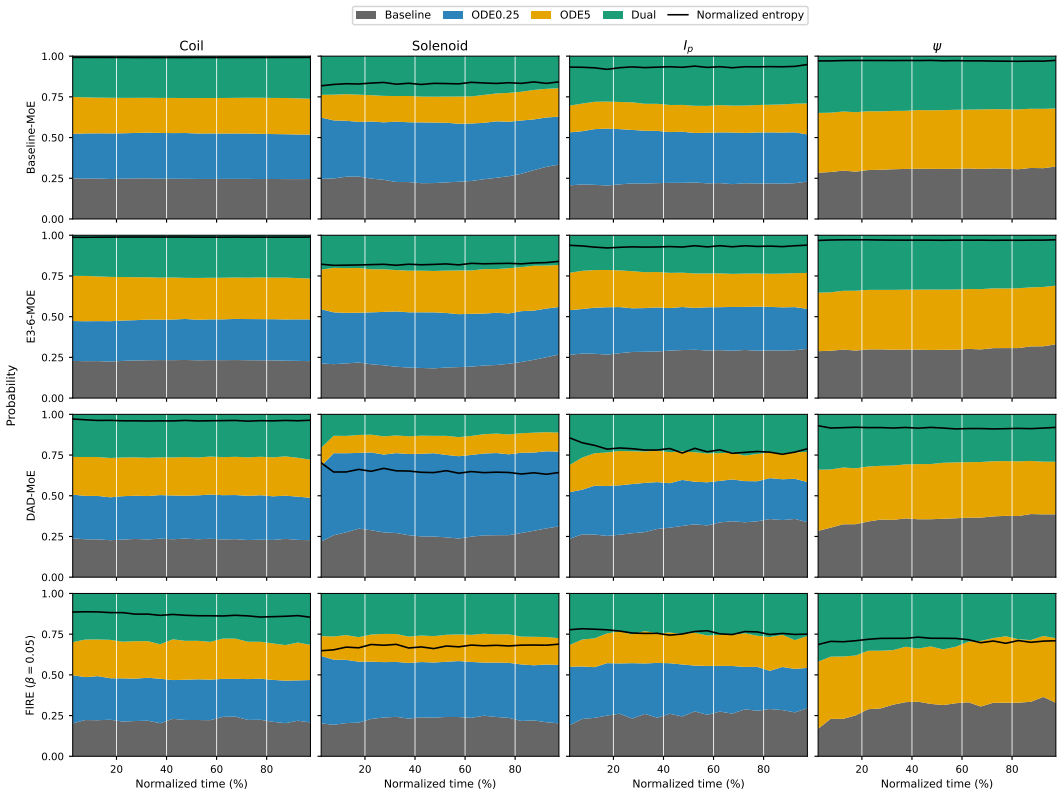

Figure 9: Validation-only routing probabilities and normalized per-head entropy for the four MAST routers over the complete recursive rollout. The rows show Baseline-MoE, E3-6-MOE, DaD-MoE, and FIRE with $\beta = 0.05$. The columns correspond to Coil, Solenoid, I_p, and ψ.

The I_p head shows a similar separation. DaD-MoE and FIRE reduce entropy to 0.785 and 0.762, compared with 0.933 and 0.931 for Baseline-MoE and PPO-MoE. FIRE achieves the lowest I_p error in Table 2, followed by DaD-MoE. The ψ routing distributions are more similar across methods, especially for Baseline-MoE and PPO-MoE, whose entropies remain high at 0.972 and 0.970. DaD-MoE and FIRE reduce these values to 0.916 and 0.713. The smaller error differences for ψ suggest that stronger routing concentration does not necessarily translate into larger gains when the available expert differences are limited.

Across both datasets, the entropy trajectories reveal that the routers do not learn a single global preference over experts. Instead, they develop head-specific routing strategies that evolve during recursive prediction. These patterns indicate that the MoE framework is exploiting complementary expert behaviors across physical variables rather than simply selecting experts according to their average one-step accuracy. The relationship between routing specialization and forecasting improvement further suggests that expert selection is most effective when candidate predictors provide distinct advantages for particular physical dynamics.

A.7 Expert-bank Transfer and Data Exposure Analysis

We study two factors that may affect routing performance beyond the main experiments: changes in the expert bank and the composition of router training data. The first study examines whether a learned routing policy can transfer to a new expert architecture without retraining. The second study separates the effect of expert exposure from the amount of data available for learning the router. All evaluations use the same 664-shot formal-test cohort and report pooled complete-rollout Macro NRMSE.

Table 12: Three-seed pooled complete-rollout Macro NRMSE for the FIRE data-exposure conditions. Values are means with population standard deviations.

Condition	Expert-pool relation	Exposure protocol	Macro NRMSE
S3_DATA	seen, matched-size subset	1,992 shots	0.06655 ± 0.00022
U3	unseen router-training pool	1,992 shots	0.06649 ± 0.00038
S5	seen expert-training pool	3,321 shots	0.06536 ± 0.00026
S5U3	seen plus unseen pools	5,313 shots	0.06543 ± 0.00040
U3-C2.67	unseen router-training pool	2.667-fold exposure	0.06718 ± 0.00136
S5U3-SE	seen plus unseen pools	matched exposure	0.06528 ± 0.00026

Transfer to new expert architectures. A routing policy is learned from the predictive characteristics of its expert bank, and it is unclear whether the learned selection strategy can be directly transferred when the underlying predictors change. We evaluate this question using Baseline-MoE, PPO-MoE, DaD-MoE, and FIRE without retraining the routers. The original router checkpoints are applied to four newly trained FNO experts on the formal-test split containing 664 shots and 16,200 valid forecast positions. We consider two transfer settings. In FNO_FULL, both the policy bank and execution bank are replaced by the FNO experts. In FNO_ACTION, the router continues to use the original TF4 experts while the execution stage uses the FNO experts. Probability-ensemble evaluation gives pooled complete-rollout Macro NRMSE values of 0.09779 for FNO_FULL and 0.13615 for FNO_ACTION, compared with 0.06506 for the original FIRE evaluation. These values correspond to increases of 50.31% and 109.26%, respectively. Figure 10 visualizes the two transfer settings and their performance changes.

The degradation under both transfer settings indicates that routing performance is closely tied to the expert representations available during training. The router does not simply learn an architecture-independent preference over predictors. Instead, effective routing depends on the predictive characteristics of the expert bank, suggesting that modifications to the expert composition should be evaluated together with the routing policy.

Effect of router training data exposure. We next examine how the source and amount of router training data influence routing performance. The goal is to separate two factors that are often coupled in practice: whether router trajectories are seen by the experts and how much data are provided for learning the routing policy. The frozen experts are trained on $E5$, which contains 3,321 expert-training shots, while $R3$ contains 1,992 router-training shots that are unseen by the experts. We compare several training pools. S3_DATA is a 1,992-shot subset of $E5$, U3 is $R3$, S5 is the complete $E5$ pool, and S5U3 is the union of $E5$ and $R3$. All conditions use the same frozen four-expert bank and three router seeds.

At matched data size, S3_DATA and U3 achieve similar mean pooled complete-rollout Macro NRMSE values of 0.06655 ± 0.00022 and 0.06649 ± 0.00038, respectively. This indicates that expert exposure alone does not determine routing performance when the amount of router data is fixed. Increasing the expert-seen pool from S3_DATA to S5 reduces the score to 0.06536 ± 0.00026, corresponding to a 1.79% improvement. Adding the unseen $R3$ shots in S5U3 gives 0.06543 ± 0.00040, which does not further improve over S5. Under the fixed-exposure supplemental protocol, U3-C2.67 uses the $U3$ pool with a 2.667-fold exposure budget and reaches 0.06718 ± 0.00136, while S5U3-SE uses the combined pool with matched exposure and reaches 0.06528 ± 0.00026.

Together, these comparisons suggest that routing performance benefits more from increasing the amount of informative training data than from simply adding trajectories unseen by the expert bank. They also show that the behavior of a routing policy depends jointly on the expert bank and the data distribution used for router training. The complete comparison is summarized in Table 12 and Figure 11.

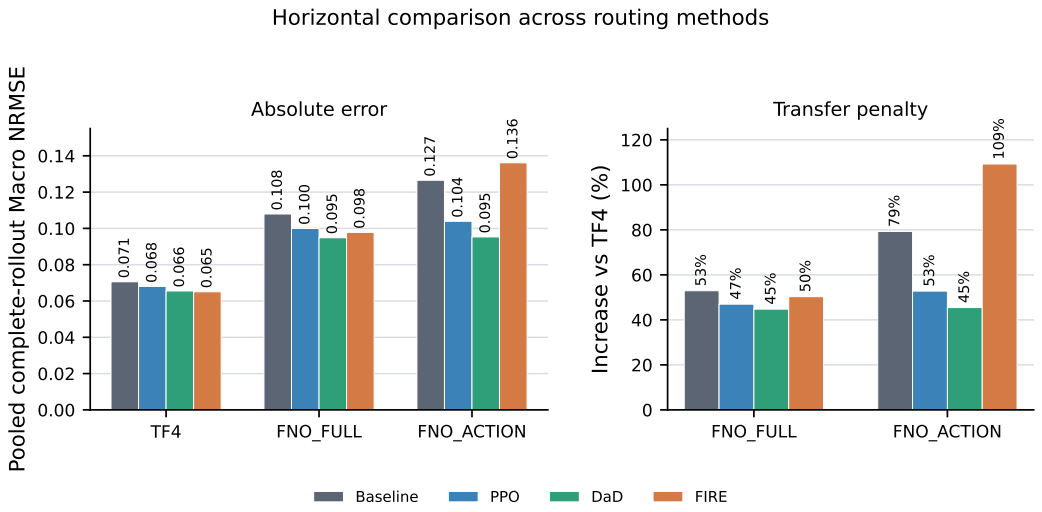

Figure 10: Horizontal comparison of Baseline-MoE, PPO-MoE, DaD-MoE, and FIRE under expert-architecture transfer. The left panel shows pooled complete-rollout Macro NRMSE on the 664-shot formal-test cohort, and the right panel shows the increase relative to each method's original TF4 result. `FNO_FULL` replaces both the policy and execution banks, whereas `FNO_ACTION` replaces only the execution bank and retains the original TF4 policy bank.

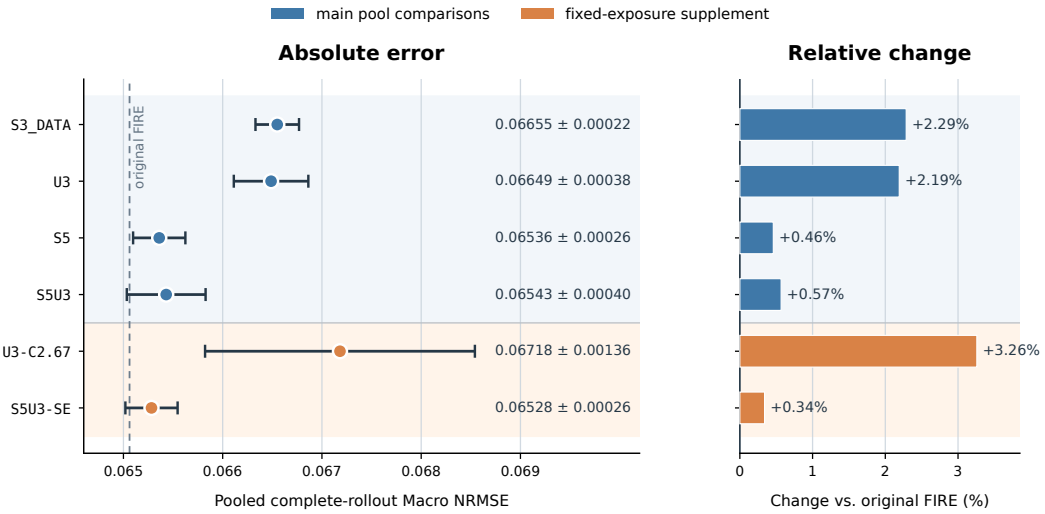

Figure 11: Seen and unseen training-pool comparisons for the frozen four-expert FIRE bank. Points show the three-seed mean pooled complete-rollout Macro NRMSE and whiskers show population standard deviation. `S3_DATA` and `U3` compare expert-seen and expert-unseen pools at matched size, while the remaining conditions vary pool composition or the exposure budget.

A.8 Additional related-work context

The main text summarizes the following two strands of prior work to keep the central narrative focused. The fuller comparisons and distinctions are retained here.

Mixtures of experts for heterogeneous time-series dynamics. Mixture-of-experts (MoE) models provide a general mechanism for conditional computation by routing inputs to specialized predictors (Jacobs et al., 1991; Shazeer et al., 2017). This idea has recently been explored in time-series

forecasting, where expert specialization can capture heterogeneous temporal structures. MoLE jointly learns linear experts and an input-dependent router (Ni et al., 2024), while AdaMixT and MoHETS combine experts operating at different temporal scales or with different inductive biases (Zhang et al., 2025; Ortigossa et al., 2026). Related online approaches adapt expert weights from sequential predictive feedback: MoE-F updates the contribution of pretrained experts according to their recent performance, and MOOE combines fixed offline experts with an adaptive online expert under distribution shift (Saqur et al., 2025; Zhao et al., 2025). Multiple-model prediction has also been studied in learned dynamics; for example, trajectory-wise multiple-choice learning selects among prediction heads based on recent transition errors (Seo et al., 2020). These methods demonstrate the value of expert specialization, but typically optimize routing with local prediction objectives or short-term performance signals. Our setting differs from existing MoE forecasting methods in three ways. First, the router learns how to effectively select among an existing and heterogeneous model bank rather than jointly optimizing the experts and the router. Second, routing operates not only along the temporal dimension but also across groups of coupled physical variables, allowing the router to exploit specialization across both dynamical regimes and physical variables. Third, the selected experts' outputs at each step determines the inputs used in subsequent recursive steps, so routing decisions must account for their effects on the entire rollout rather than only immediate prediction error.

Sequential routing under recursive distribution shift. Routing in a recursive dynamics model is inherently sequential: selecting an expert changes the predicted state, which in turn changes the inputs and routing decisions encountered at subsequent steps. This creates a distribution shift between one-step supervision on observed trajectories and deployment on model-predicted trajectories, closely related to the exposure-bias problem studied in sequential prediction and imitation learning. Scheduled sampling trains predictors on a mixture of observed and predicted inputs (Bengio et al., 2015), while DAgger and Data as Demonstrator explicitly aggregate supervision on learner-induced states (Ross et al., 2011; Venkatraman et al., 2015). Recent interactive and hybrid imitation-learning methods further study when state-wise corrective feedback improves over pure behavior cloning (Li & Zhang, 2025). Beyond local corrective supervision, AggreVaTe incorporates cost-to-go information into sequential decisions (Ross & Bagnell, 2014; Sun et al., 2017), and policy-optimization methods directly optimize cumulative trajectory-level objectives (Peters et al., 2010; Schulman et al., 2015; 2017; Abdolmaleki et al., 2018). The importance of trajectory-level evaluation is also emphasized in recent model-based offline reinforcement learning, where model errors can compound substantially over long rollouts (Lin et al., 2026). Motivated by these observations, we treat expert routing as a sequential decision problem whose quality should be measured by the resulting multistep rollout rather than only by immediate prediction error. This perspective connects conditional expert selection with long-horizon learned dynamics: the router operates over an existing model bank, receives feedback from recursive trajectories, and is optimized for the accuracy of the resulting coupled physical variables rollout.

A.9 Limitations and future directions

Our evaluation focuses on MAST-U VDE simulations and experimental data from the MAST device. The experiments use a fixed forecast horizon and assume that future control inputs are available. The generalization of the routing framework to other devices and operating conditions remains to be evaluated. Future work can extend the study to additional devices and design targeted experiments to cover broader operating regimes. The predict-first strategy in Wang et al. (2025a) provides one possible approach for selecting informative experiments.

The current study evaluates the MoE methods and FIRE in an open-loop forecasting setting. The router is not integrated with a controller, and the training objective does not explicitly consider actuator limits, stability constraints, or prediction uncertainty. Future work can investigate closed-loop control scenarios and incorporate uncertainty estimation and safety-related constraints. These extensions can build on recent progress in robust plasma trajectory design, controller assessment, and full-horizon offline learning (Wang et al., 2025a; Lvovskiy et al., 2025; Lin et al., 2026).