SUPPORT-AWARE SHIELDING RESTORES SAFETY FOR PREDICTIVE WORLD-MODEL CONTROL UNDER SHIFT

Anonymous authors
Paper under double-blind review

ABSTRACT

A predictive world model is supposed to let an operator ask "what if" before committing to a change, and that promise rests entirely on the model knowing when it is wrong. We study what becomes of the promise when the world moves. Working in a setting where every planning decision can be replayed against a full-information oracle, we find a pattern that repeats across every model we train: the planners that do best are the planners that are most dangerous. The models that extract the most benefit under familiar conditions are precisely the ones that breach the operator's safety constraints once a crowd forms, and the uncertainty intervals a shield would have to trust quietly stop covering the truth exactly when they are needed. This failure is invisible from forecasting accuracy alone: models that are nearly indistinguishable on held-out prediction error behave entirely differently once their forecasts are allowed to drive actions. We then show that the remedy is not a better forecaster. It is a planner that keeps asking whether it still recognises the world. A controller that re-plans as it goes, replays its own model against what it has just observed, checks whether its forecast has wandered outside the region its training data covered, and declines to act when either test fails, recovers nearly all of the lost safety while surrendering only a small and bounded share of the achievable benefit. We prove two statements that explain the exchange - one tying how reliably shifted situations are recognised to how often constraints can be breached, one showing that the cost of declining to act is paid in proportion to how often it is declined - and we machine-check both. The lesson we draw is that safety under distribution shift is more cheaply bought with an online admission of ignorance than with a more confident model.

1 INTRODUCTION

Learned world models promise a way to act on a system without experimenting on it. If a model can predict what a network, a plant, or a fleet would do under an intervention it has never seen, an operator can search over interventions offline and deploy only the one that wins (Ha & Schmidhuber, 2018; Hafner et al., 2019; 2025). In any setting where the system is live, that search is not run for reward alone: it is run subject to constraints, and the usual way to enforce a constraint against a learned model is to make the model report an uncertainty interval and refuse any action whose pessimistic end violates the constraint (Alshiekh et al., 2018; Berkenkamp et al., 2017; Chua et al., 2018).

This arrangement has a soft joint. The shield is only as trustworthy as the interval, and intervals are calibrated on the training distribution. There is by now substantial evidence that predictive uncertainty degrades under dataset shift (Ovadia et al., 2019; Nalisnick et al., 2019), and a literature on restoring coverage when the shift can be characterised in advance (Tibshirani et al., 2019; Gibbs & Candès, 2021; Barber et al., 2023). What is less clear is the operational question: when the intervals degrade, what actually happens to a planner that was built on top of them, and what is the cheapest thing one can do about it?

We study that question concretely. Our testbed is traffic steering in a multi-cell mobile network: at each decision step a controller may retune admission and handover parameters per cell, it is scored

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.

on a weighted basket of quality indicators, and it is subject to hard baseline-relative constraints on the two indicators an operator would never let regress - the call drop rate and the inter-cell handover success rate. The setting has two properties that make it unusually informative. First, a simulator lets us compute the *oracle* plan - the same search procedure run against the true system instead of the learned model - so a planner's quality is a regret, not a raw score. Second, because interventions can be replayed under common random numbers (Glasserman & Yao, 1992), counterfactual comparisons are paired and therefore low-variance.

What we find. Across twelve world models, three training seeds, and two shift regimes, three things hold consistently.

(i) There is a safety-fidelity frontier, and it is tight. In distribution, how much regret a model removes correlates strongly and negatively with how often it violates a safety constraint (Figure 1a; Pearson $r = -0.84$, $p = 0.005$). Models beat the oracle's caution only by taking actions the oracle declined, and those are exactly the unsafe ones. No model occupies the "accurate and safe" corner.

(ii) Calibration fails in two stages, and accuracy does not warn you. Re-parameterising the demand surge costs about ten points of interval coverage; replacing the environment model, with the feature interface unchanged, costs another twenty-five, leaving intervals that cover the truth roughly half the time (Figure 3). Under the first regime every open-loop planner becomes unsafe on a third to over half of held-out scenarios, though their held-out prediction errors differ by at most a fifth.

(iii) Re-checking support online restores safety cheaply. A planner that re-plans hourly and abstains - reverting to the baseline - when either a self-consistency replay or a predicted-state support test fails reduces violations from eleven-sixteen out of thirty scenarios to zero-one, while holding regret to roughly one utility point against a closed-loop oracle (Figure 4). It abstains on about a third of steps, and we show this is very nearly the entire cost.

We complement the measurements with two elementary bounds - one relating detector recall to the realised violation rate, one pricing abstention - and machine-check both in Lean 4 (de Moura & Ullrich, 2021; The mathlib Community, 2020), with the proof scripts included in the supplement.

A deliberate non-result. Part of our model zoo studies sigmoid versus softmax gating in a mixture of predictors, a comparison for which there is recent theoretical support (Nguyen et al., 2024). We find that sigmoid gating changes the *mechanism* - it is the only gate under which several experts are genuinely active at once, whereas softmax gates degenerate to a winner-take-all that never clears threshold (Figure 10) - and that it is the safest of the models willing to act in distribution. We do *not* find that it is more accurate: on our flagship closed-loop comparison the paired difference against softmax has a 95% confidence interval of $[-0.0010, +0.0015]$, a null. We report it as such.

Contributions. (1) A paired-counterfactual benchmark for safety-constrained world-model control with an oracle-regret protocol, a fail-closed feature contract, and two independently implemented environments (Section 3). (2) Evidence for a safety-fidelity frontier and for two-stage calibration collapse, neither of which is visible from forecasting metrics (Sections 6.2 and 6.3). (3) A closed-loop planner whose two support checks restore safety under shift, with machine-checked bounds on what it guarantees and what it costs (Sections 4.3 and 5). (4) A mechanism audit separating what sigmoid gating does from what it does not do, reported with its null result intact (Section 6.6).

2 Related Work and Background

Our predictor follows the joint-embedding predictive template (LeCun, 2022; Assran et al., 2023; 2025), with an exponential-moving-average target (Grill et al., 2020) and an anti-collapse term (Bardes et al., 2022), and is used for planning in the manner of learned world models (Ha & Schmidhuber, 2018; Hafner et al., 2019; 2025). The difference that matters here is that its latent must support *calibrated interval* heads, because a shield consumes quantiles rather than point predictions. Our predicted-state check is a relative of the pessimism used in model-based and offline RL (Chua et al., 2018; Yu et al., 2020; Kidambi et al., 2020), but it is applied at decision time in closed loop and yields a binary abstention rather than a reward penalty; the shield itself follows the shielding (Alshiekh et al., 2018; García & Fernández, 2015) and constrained-MPC traditions (Mayne et al., 2000;

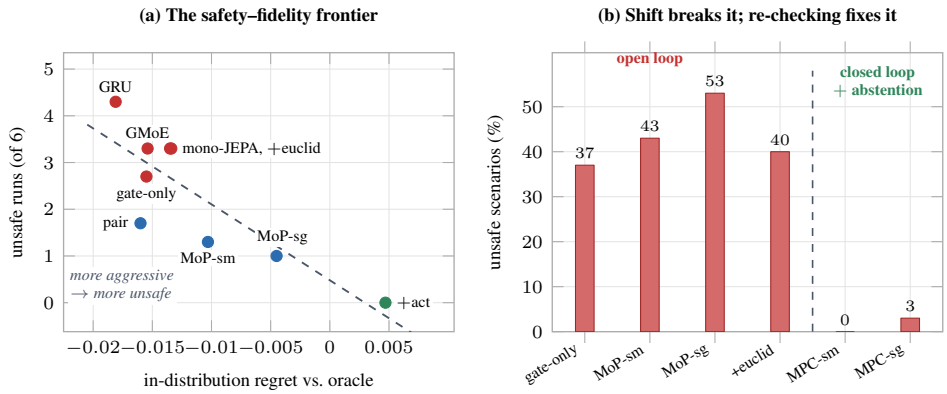


Figure 1: **The problem and the fix.** **(a)** On in-distribution scenarios, world models that cut the most regret against a full-information oracle are also the models that violate safety-critical KPIs most often (Pearson $r = -0.84$, $p = 0.005$; dashed line is the least-squares fit over the nine learned planners). No model sits in the bottom-left "accurate and safe" corner. **(b)** Under FlashCrowd shift every open-loop planner becomes unsafe on a large fraction of held-out scenarios; re-planning in closed loop with a support-aware abstention rule returns the violation rate to near zero on the same scenario distribution. Percentages in (b) are over 30 open-loop and 30 closed-loop seeds. In (a) the monolithic JEPA and the Euclidean-gate variant coincide to three decimal places and share one label.

Rawlings et al., 2017). That predictive uncertainty degrades under shift is well established (Ovadia et al., 2019; Quiñonero-Candela et al., 2009; Koh et al., 2021), as are conformal repairs when the shift is characterisable (Tibshirani et al., 2019; Gibbs & Candès, 2021; Barber et al., 2023) and the long tradition of abstention (Chow, 1970; Geifman & El-Yaniv, 2017). What we contribute is not a new estimator in either family but the observation that, inside a control loop, a coarse support test used to *abstain* recovers more safety than the calibrated intervals themselves can supply once shifted. We use sigmoid versus softmax gating (Jacobs et al., 1991; Shazeer et al., 2017; Fedus et al., 2022; Nguyen et al., 2024) as a controlled mechanism probe rather than a scaling device. Section A gives the full discussion, and Section B the network-control background (Polese et al., 2023; Patriciello et al., 2019; Iversen, 2010).

3 Problem Setup and Benchmark

Decision problem. A cell site is a set of n cells observed in five-minute bins. At bin t each cell i exposes counters (setup requests, connected users, admission attempts and rejects), a current action $a_{i,t} = (U, \mathrm{HOM}, \mathrm{TTT})$ - an admission ceiling, a handover margin and a time-to-trigger - and five quality indicators reported as event/exposure ratios: setup success, random-access success, call drop rate, and inter- and intra-cell secondary-cell success. Drop rate is lower-is-better; the other four are higher-is-better.

A *plan* assigns actions to cells over a horizon. Its value is the weighted gain in those indicators, plus connectivity, relative to a fixed baseline schedule A_0 (weights in Section C). Two indicators - drop rate and inter-cell handover success - are additionally *safety-critical*: a plan is unsafe if its worst-case-over-time value on either regresses against A_0 by more than a tolerance. We report the **regret** $J^\star - \hat{J}$ of a planner against an *oracle* that runs the identical search against the true simulator, and the number of evaluation scenarios on which the realised plan is unsafe. Because both planners share a search procedure and differ only in their rollout function, regret isolates world-model quality for control.

Environments. Three environments emit the same interface (Figure 5). **P0-A** is an analytic digital twin: Erlang-B admission with retry feedback, load-difference handover spillover, and indicator rates driven by blocking, saturation, retry intensity and handover stress. It supplies all training data and all in-distribution and FlashCrowd-shift evaluation. **P0-C** is an independently written simulator

with deliberately different mechanics - weekly-harmonic demand with autoregressive lognormal noise, a Gaussian-approximated blocking solver, three-bin geometric retries, hysteretic handover flows, and multiplicative rather than additive indicator noise - used only for zero-shot transfer. **P0-B** instantiates the interface on ns-3 with the 5G-LENA NR stack (Riley & Henderson, 2010; Patriciello et al., 2019) at 3 and 28 cells. It is a scale and interface check and contributes no training data and no model comparison; Section 7 states plainly what it does and does not show.

Feature contract, pairing, and protocol. Models see 30 features per cell per bin over a 36-bin history, and a fail-closed allowlist rejects every generator-internal variable - offered load, blocking probability, crowd multiplier, mechanism labels - so a model cannot read the answer (Section D). Every counterfactual rollout of a scenario reuses that scenario's exogenous randomness, so two schedules differ only through the intervention (Glasserman & Yao, 1992); both properties are unit-tested (Section R). Twelve models span three structured baselines and nine neural predictors, listed in full in Section H. Training uses twelve scenarios of four simulated days; evaluation uses six held-out scenarios in distribution and thirty under shift, with means over three training seeds and 95% bootstrap intervals from 5000 resamples (Efron, 1979). Hyperparameters are in Section G and seeds in Section I.

4 Method

4.1 Sigmoid Mixture-of-Predictors JEPA

The predictor (Figure 7) maps a history and a candidate action sequence to a latent from which interval-valued indicator forecasts are decoded. A GRU encoder (Cho et al., 2014) summarises each cell's history as $\mathbf{z}_t$; a query network fuses it with an action embedding, a cell embedding and an hour-of-day embedding to give $\mathbf{q}$.

Mixture of residual predictors. Five experts are each tied to a named mechanism and restricted to a subset of features and a look-back window: demand and seasonality (36 bins), access pressure (6), mobility (12), admission residual (12), and retry congestion (6); the exact assignments are in Section E. Writing ΔP_r for expert r and P_0 for a shared base predictor,

$$\hat{\mathbf{z}} \;=\; \mathrm{LayerNorm}\Big(P_0(\mathbf{q}) \;+\; \tfrac{\alpha}{\sqrt{R}} \textstyle\sum_{r=1}^{R} g_r \, \Delta P_r(\mathbf{q}, \mathbf{c}_r)\Big), \qquad g_r = \sigma\big(s_r/\tau\big), \tag{1}$$

with $R = 5$, $\alpha = 1$, $\tau = 0.5$. The gates are *independent* sigmoids, so $\sum_r g_r$ is unconstrained and any subset of mechanisms - including all or none - may be active. The softmax ablation replaces g with $\mathrm{softmax}(s/\tau)$; the gate-only ablation deletes P_0 so the experts must carry the whole prediction; the Euclidean variant replaces the inner-product score s_r with a dimension-normalised radial-basis score, which removes the temperature-score interaction. Gate biases are initialised at -1 to avoid an all-on start.

Heads and losses. Four heads read [$\hat{\mathbf{z}}$, action embedding], keeping the action visible to the decoder: event and exposure counts under negative-binomial likelihoods (Cameron & Trivedi, 2013), state counts under a Poisson likelihood, and five quantiles $\{0.05, 0.1, 0.5, 0.9, 0.95\}$ of each indicator ratio under the pinball loss (Koenker & Bassett, 1978). The joint-embedding term is a mean-squared error between ℓ_2-normalised $\hat{\mathbf{z}}$ and an exponential-moving-average target encoder, with a variance floor preventing collapse (Bardes et al., 2022; Grill et al., 2020). Two optional action-sensitivity terms define two further ablations. All loss terms, weights and the specialisation regulariser are given in Section F.

4.2 Shielded Greedy Planner

Both the oracle and every learned planner run one procedure and differ only in the rollout function. Starting from the baseline schedule, the planner repeatedly proposes a single cell-action change from a small grid, rolls it out, and keeps the best improving candidate, for at most $K = 4$ changes. A candidate is admissible only if it clears the **shield**: its conservative indicator trajectory must not regress the baseline on either safety-critical indicator by more than a margin. Conservative here means the model's own tail quantiles - the q_{95} of the drop rate and the q_{05} of inter-cell handover

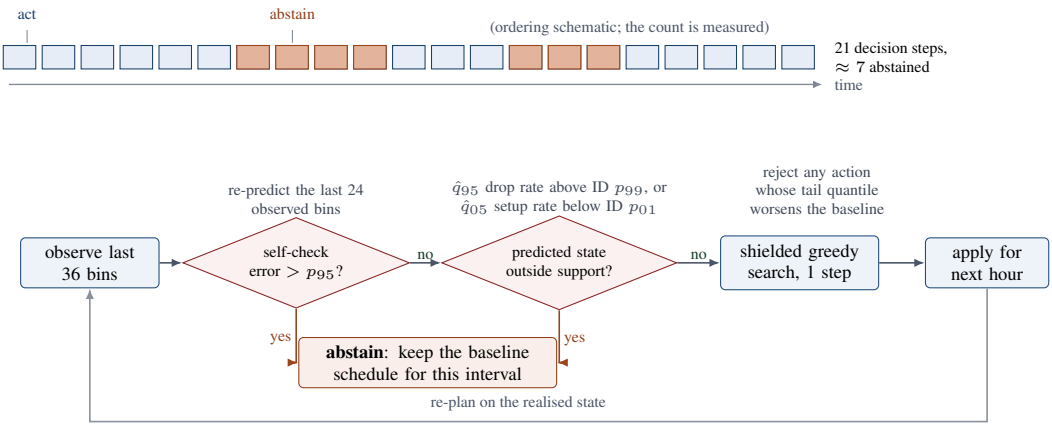


Figure 2: **Closed-loop control with support-aware abstention.** Every hour the planner re-reads the realised state and asks two independent questions before it is allowed to act. The *self-check* replays the model against the 24 bins it has just observed and compares the error to a threshold calibrated on in-distribution data; the *predicted-state check* rolls the baseline forward and asks whether the model's own conservative forecast leaves the region the training data covered. Failing either question, the planner abstains and keeps the baseline schedule — an action that is safe relative to the baseline by construction. Only when both checks pass does the quantile shield get to choose an intervention.

success - so the shield reads exactly the quantities whose calibration Section 6.3 shows to be fragile. The oracle supplies true trajectories in both places.

4.3 Closed-Loop Control with Support-Aware Abstention

The open-loop planner commits to one plan from one forecast. Our controller (Figure 2) instead re-plans every twelve bins on the realised state, and before it is permitted to act it must pass two support tests.

Self-check. The model re-predicts the most recent 24 observed bins from the preceding history and its own realised actions. The mean absolute indicator error is compared against a threshold calibrated as the 95th percentile of the same statistic on in-distribution training windows. This asks whether the model still explains what just happened.

Predicted-state check. The model rolls the baseline schedule forward over the remaining horizon. If its q_{95} drop rate exceeds the in-distribution 99th percentile, or its q_{05} setup success falls below the in-distribution 1st percentile (each with a small slack), the forecast has left the region the training data covers. This asks whether the model is being asked about a world it has seen.

Failing either test, the controller **abstains**: it keeps the baseline schedule for that interval. Abstention is safe relative to the baseline by construction, since safety is defined as a baseline-relative regression. When both tests pass, the shielded greedy step of Section 4.2 runs on the remaining horizon. The closed-loop oracle runs the same loop with true rollouts and never abstains, so regret continues to measure only what the learned model costs. Thresholds are fixed before any shifted scenario is seen.

5 What the Shield Guarantees, and What It Costs

Two elementary statements explain the exchange the controller makes. Both condition on the type of a decision step and are therefore assumption-light; what they are not is a coverage guarantee, and Section 7 says so.

Assumption 1 (Step decomposition)**.** A fraction π_{OOD} of decision steps are out-of-support. On in-support steps the shield's upper quantile is calibrated to within $\varepsilon_{\mathrm{cal}}$, so the per-step violation probability there is at most $0.05 + \varepsilon_{\mathrm{cal}}$. The support detector has recall β on out-of-support steps.

Abstention reproduces the baseline schedule, whose baseline-relative violation margin is zero by construction.

Proposition 1 (Realised violation rate)**.** *Under Assumption 1, the realised violation rate obeys*

$$V \;\le\; (1-\pi_{\mathrm{OOD}})\,(0.05+\varepsilon_{\mathrm{cal}}) \;+\; \pi_{\mathrm{OOD}}\,(1-\beta). \tag{2}$$

In particular, if $\beta \to 1$ then $V \le 0.05 + \varepsilon_{\mathrm{cal}}$ irrespective of how much of the workload has shifted.

Proposition 2 (Price of safety)**.** *Let π_{abs} be the fraction of steps on which the controller abstains, $\bar{U}_{\max}$ the largest utility gain a single step can contribute, and Δ_{insup} the model-versus-oracle gap on steps where it does act. Then*

$$\mathrm{Regret} \;\le\; \pi_{\mathrm{abs}}\,\bar{U}_{\max} \;+\; (1-\pi_{\mathrm{abs}})\,\Delta_{\mathrm{insup}}. \tag{3}$$

The proofs condition on step type and bound each branch: a missed shifted step is bounded only trivially, a flagged one contributes nothing, and an abstained step forfeits at most one step's gain. Both statements, their limiting corollaries, and numerical instantiations at the measured operating point are machine-checked in Lean 4 with Mathlib (de Moura & Ullrich, 2021; The mathlib Community, 2020); the development is in the supplement and Section Q, and Figure 12 plots the bounds against the measurements.

Two readings are worth recording, and both run the bounds *backwards* from measured quantities rather than plugging in assumed constants. Treating the shifted evaluation as entirely out-of-support - the pessimistic case - Equation (2) says the observed one violation in thirty scenarios requires a detector recall of at least $1 - 1/30 \approx 0.97$. On the cost side, the controller abstains on about 7 of 21 steps, so Equation (3) with $\pi_{\mathrm{abs}} \approx 1/3$ attributes the measured regret of $+0.011$ entirely to abstention as soon as a single step can contribute about 0.033 of utility, leaving $\Delta_{\mathrm{insup}} \approx 0$. In other words the world model is not the binding constraint at this operating point: what regret remains is the price of declining to act, not of predicting badly.

6 Experiments

We answer five questions: do the models forecast well (Section 6.1); is there a safety-fidelity frontier (Section 6.2); what does shift do to calibration (Section 6.3); does online re-checking help (Section 6.4); and what does the gate actually change (Section 6.6). Counterfactual-fidelity results, per-seed tables and the ns-3 instantiation are in Sections J, K and P.

6.1 Forecasting and Calibration In Distribution

Table 1 (left block) gives held-out indicator error and interval coverage. The neural predictors improve on the best structured baseline by 1.6-1.9× in ratio error and cluster tightly: excluding the action-decoder ablation, the remaining eight sit between 0.0195 and 0.0234, with three-seed standard deviations of 0.0004-0.0015. Coverage is close to the nominal 0.90 everywhere; every other recorded metric is in Section L. The action-decoder variant is the one clear regression - its auxiliary objective competes with the quantile head, costing both accuracy and rare-event recall - and we keep it in the zoo because it turns out to be the safest model in distribution, for reasons that have nothing to do with accuracy.

The point is negative and important: *these numbers do not order the models the way control does*. Ratio errors within 20% of one another correspond to violation counts differing by more than $4\times$.

6.2 The Safety-Fidelity Frontier

Table 1 (right block) reports oracle regret and violation counts on six held-out in-distribution scenarios. Negative regret means a planner extracted *more* utility than the oracle's own shielded search. That sounds like a win and is not: the oracle declines actions its true rollout shows to be unsafe, so a learned planner can only beat it by taking actions the oracle refused. Exactly as that reasoning predicts, the ranking by regret is close to the reverse of the ranking by safety (Figure 1a; Pearson $r = -0.84$, $p = 0.005$; Spearman $\rho = -0.71$, $p = 0.031$). The monolithic recurrent model posts the most negative regret, -0.0181, and is unsafe on 4.3 of 6 runs; the action-decoder variant is unsafe on none, because it is too pessimistic to act at all.

Table 1: **In-distribution results: accuracy does not predict safety.** Left: held-out one-step forecasting. Right: open-loop planning on six held-out scenarios against a full-information oracle running the identical search. Negative regret means the planner beat the oracle, which it can only do by taking actions the oracle's shield refused. Means over three training seeds (± s.d. across seeds); intervals are 95% bootstrap (5000 resamples, Section I). Best per column in **bold**.

	Forecasting		Control (open loop)		
Model	ratio MAE ↓	coverage	regret ↓	95% CI	unsafe /6 ↓
Persistence	0.0443 ±0.0048	–	–	–	–
Seasonal	0.0443 ±0.0048	–	–	–	–
GBDT + Erlang-B	0.0368 ±0.0043	–	–	–	–
GRU (monolithic)	0.0234 ±0.0015	0.897	−0.0181 ±0.0125	[−0.034, −0.004]	4.3
Graph MoE (softmax)	0.0196 ±0.0009	0.890	−0.0154 ±0.0027	[−0.019, −0.012]	3.3
Gate-only sigmoid MoE	0.0200 ±0.0005	0.889	−0.0155 ±0.0045	[−0.022, −0.011]	2.7
Monolithic JEPA	0.0228 ±0.0014	0.893	−0.0134 ±0.0188	[−0.037, +0.008]	3.3
MoP-JEPA (softmax)	**0.0195** ±0.0007	0.893	−0.0103 ±0.0086	[−0.018, +0.002]	1.3
MoP-JEPA (sigmoid)	0.0201 ±0.0014	0.890	−0.0045 ±0.0081	[−0.016, +0.003]	**1.0**†
+ action decoder	0.0351 ±0.0014	0.909	+0.0047 ±0.0023	[+0.002, +0.008]	0.0‡
+ paired-CF loss	0.0205 ±0.0004	0.895	−0.0160 ±0.0098	[−0.028, −0.004]	1.7
+ Euclidean gate	0.0202 ±0.0007	0.897	−0.0135 ±0.0013	[−0.015, −0.012]	3.3

† safest among models that intervene. ‡ safe by inaction: this variant almost never passes its own shield, so it takes no action and inherits the baseline.

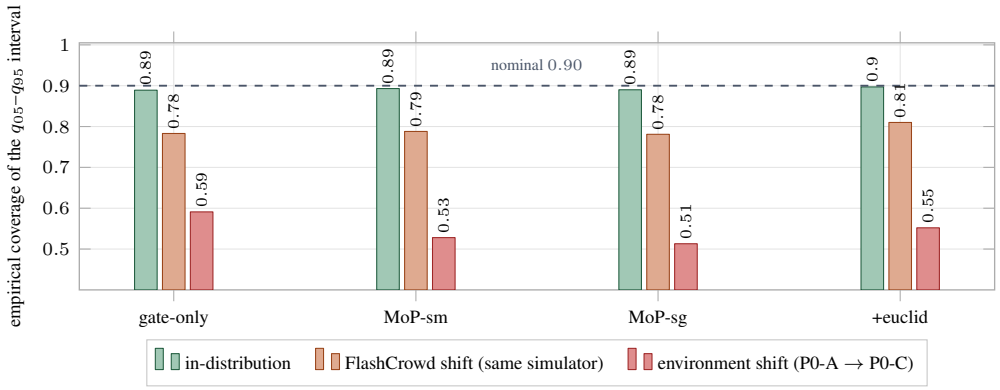


Figure 3: **Calibration degrades in two stages.** Re-parameterising the FlashCrowd events costs about ten points of coverage while leaving the environment model untouched; replacing the environment model itself, feature contract fixed, costs another twenty-five. Safety shields read exactly these quantiles, which is why the open-loop planners of Table 2 fail as they do. In-distribution values are three-seed means, the shifted regimes six-seed runs (Sections N and O).

Within the models that do act, the sigmoid-gated mixture is the safest (1.0/6) at a regret indistinguishable from zero, against 2.7/6 for the gate-only sigmoid mixture and 3.3/6 for the softmax graph mixture. We regard this as the most useful comparison in the table, and we note its limits: six scenarios and three seeds cannot separate 1.0 from 1.3.

6.3 Calibration Collapses in Two Stages

Figure 3 separates two kinds of shift that are often conflated. Re-parameterising the FlashCrowd events (Section B.3) while leaving the environment model untouched costs roughly ten points of coverage, from ≈ 0.89 to 0.78-0.81; swapping the environment model for the independently written P0-C, feature contract unchanged, costs a further twenty-five, to 0.51-0.59. This is the mechanism behind every open-loop failure that follows: the shield reads q_{95} and q_{05}, and when the interval

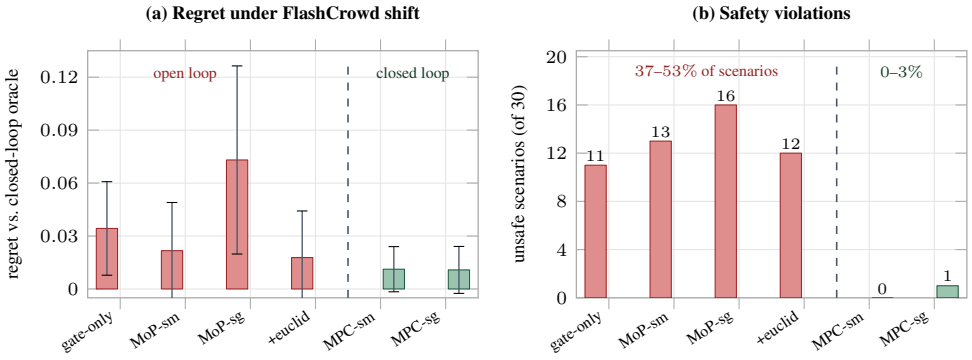


Figure 4: **Out-of-distribution control, 30 held-out FlashCrowd scenarios.** Open-loop planners commit to a whole-horizon plan from one forecast; closed-loop planners re-plan hourly and may abstain. Error bars are one standard deviation across scenarios. Re-checking support cuts regret and violations at once, without ever making the planner more aggressive: every intervention still has to clear the same quantile shield.

Table 2: **Under FlashCrowd shift, re-checking support online is what restores safety.** Thirty held-out shifted scenarios. Models and abstention thresholds are fixed on in-distribution data before any shifted scenario is seen. Both rows are scored against an oracle operating in the same regime, so the closed-loop rows are not flattered by re-planning alone. Intervals are 95% bootstrap over scenarios.

Model	regret ↓	95% CI	unsafe /30 ↓	abstained /21
Open loop: one plan per scenario				
Gate-only sigmoid MoE	$+0.0343 \pm 0.0265$	–	11	–
MoP-JEPA (softmax)	$+0.0217 \pm 0.0273$	–	13	–
MoP-JEPA (sigmoid)	$+0.0731 \pm 0.0533$	–	16	–
+ Euclidean gate	$+0.0178 \pm 0.0264$	–	12	–
Closed loop, re-plan hourly, support-aware abstention (ours)				
MoP-JEPA (softmax)	$+0.0112 \pm 0.0128$	$[+0.0070, +0.0161]$	**0**	7.1
MoP-JEPA (sigmoid)	$+0.0108 \pm 0.0133$	$[+0.0066, +0.0160]$	**1**	6.9
Paired difference (softmax − sigmoid), closed loop				
difference	$+0.0003 \pm 0.0035$	$[-0.0010, +0.0015]$	null; 18/30 seeds favour softmax	

covers the truth half the time the shield is not conservative, it is merely wrong, and it passes actions it should have blocked.

6.4 Online Re-Checking Restores Safety

Table 2 and Figure 4 give the main result. Under FlashCrowd shift, open-loop planners are unsafe on 11-16 of 30 scenarios, and their regret turns sharply positive - they are now worse than the oracle as well as unsafe. The closed-loop controller with support-aware abstention is unsafe on $0/30$ and $1/30$ for the softmax and sigmoid mixtures respectively, with bootstrap intervals on regret of $[+0.0070, +0.0161]$ and $[+0.0066, +0.0160]$. It abstains on 6.9-7.1 of 21 decision steps.

Two controls matter. The comparison is against a *closed-loop* oracle that also re-plans, so the gain is not an artefact of re-planning; and abstention cannot manufacture utility, so the regret it buys safety with is bounded by Proposition 2, and the measured value sits at that bound. The paired difference between the two gated models is $+0.0003 \pm 0.0035$, interval $[-0.0010, +0.0015]$, with softmax winning 18 of 30 scenarios: no accuracy advantage to sigmoid gating, and we claim none.

6.5 Environment Shift, and a Reversal

Table 12 evaluates models trained on P0-A zero-shot on P0-C. Forecast error roughly triples, coverage collapses, and the regret ranking *inverts*: the softmax and sigmoid mixtures post the best (most negative) regret and are unsafe on $6/6$ and $5/6$ scenarios, while the Euclidean-score variant posts positive regret and is unsafe on $0/6$. Under environment shift, as in distribution, apparent control performance is bought with violations - a practitioner reading only the regret column would deploy the most dangerous model available.

6.6 What the Gate Changes

Figure 10 audits gate activations over three seeds. Softmax-gated mixtures never place more than 0.5 mass on any expert, so no gate clears threshold on any input and the mixture degenerates to its shared base. Sigmoid gates keep 1.5-2.2 of five experts active; the paired-counterfactual variant never switches all gates off. The Euclidean-score variant, despite a sigmoid squashing function, saturates low and behaves like the softmax models - concurrency is a property of the score as much as of the nonlinearity. So sigmoid gating delivers the mechanism it promises, and that mechanism is visible in the in-distribution safety column but not in accuracy anywhere. With the null of Section 6.4, our reading is that sigmoid gating is competitive in accuracy while enabling concurrent mechanism activation and better in-distribution safety; it is not uniformly better, and the seed spread (1.54 ± 0.68) is too large to build a claim on the activation number alone.

7 Limitations

We state the boundaries of the evidence directly; Section T gives the full form of each point.

The ns-3 environment is an interface check, not evidence. P0-B instantiates a full 5G-LENA NR stack at up to 28 cells and shows the feature contract can be populated from a running system-level deployment. But its P4 counters come from a demand model scheduled alongside the NR stack rather than from radio-level measurements, so two of the five indicators are constant by construction. No model here is trained or evaluated on P0-B data and no claim in Section 6 rests on it: this is two-environment evidence with a system-scale interface demonstration, not three-simulator evidence (Section P).

Abstention buys safety with inaction. The controller inherits whatever the baseline does; our safety criterion is baseline-relative, so a drifting baseline would not be caught. Absolute constraints need a safe fallback controller, which we do not provide.

The propositions are not coverage guarantees. They assume calibration *on in-support steps* and take the detector recall β as given; neither is verified by the theory, and β is not measured - we report the recall *consistent with* the observed violation rate. Our percentile thresholds are heuristics, not conformal predictors (Barber et al., 2023; Gibbs & Candès, 2021).

Statistical resolution. In-distribution control uses six scenarios and three seeds: enough for the frontier, not enough to separate 1.0 from 1.3 unsafe runs, which is why the flagship claim is the thirty-scenario shifted evaluation. Gate activation varies across seeds (1.54 ± 0.68), so the mechanism claim rests on the stable all-off contrast (Section K).

Sigmoid gating is not shown to be more accurate, and **the oracle is a greedy oracle**: regret is measured against the same bounded search run on the true simulator, not against a global optimum.

8 Conclusion

Predictive world models fail dangerously under distribution shift in a specific and reproducible way: the intervals a safety shield depends on stop covering the truth, the shield keeps passing actions, and the models that looked best are the ones that hurt most. Forecasting metrics do not warn you, because the models that differ enormously in safety differ marginally in accuracy. What repairs the loop is not a better forecaster but a planner that re-checks, at every decision, whether it still recognises the world - and that declines to act when it does not. The cost of that discipline is bounded, small, and almost entirely attributable to the declining itself.

IMPACT STATEMENT

This work aims to make learned controllers safer to deploy in infrastructure settings by making them decline to act when they leave the conditions they were trained for. The main positive impact is a practical, cheap mechanism for failing safe. There are two risks worth naming. First, a support test that reports "in distribution" can create unwarranted confidence; our results show that the test is useful precisely because it is conservative, and we would caution against tuning its thresholds for higher utility. Second, abstention concentrates responsibility in the fallback policy, which in a real deployment is an operational decision rather than a learned one. The application domain, mobile-network traffic steering, affects service quality rather than individual users' data; our simulators contain no personal data of any kind, and we release no models trained on operator traces.

REPRODUCIBILITY STATEMENT

All results are reproducible from the accompanying code. The full benchmark over three training seeds is driven by `scripts/vast_run_all.sh`; the thirty-seed shifted sweep by `scripts/eval_final_sweep.py`; the thirty-seed closed-loop study with bootstrap intervals by `scripts/eval_mpc_30.py`; the zero-shot environment-shift study by `scripts/eval_cross_sim.py`; and the ns-3 instantiation by `scripts/vast_ns3_28_fullnr.sh`. Nine unit tests verify simulator determinism, the fail-closed feature allowlist, common-random-number pairing across counterfactual rollouts, the no-restart action semantics, and the ns-3 trace adapter. Scenario seeds are fixed in the scripts and listed in Section I; confidence intervals use 5000 bootstrap resamples. Results were reproduced independently on Apple M1 (MPS) and on an NVIDIA RTX 3090 (CUDA) with identical conclusions; both sets of logs are included. The Lean development in Section Q builds with `lake build SafeMPC` and prints an axiom audit for every theorem.

AI USE STATEMENT

In this work, we used generative AI tools as a coding and writing assistant: for drafting and refactoring parts of the experiment-running and plotting code, for LaTeX formatting, and for copy-editing prose. We did not use generative AI tools to generate research ideas, to design the benchmark or the method, to produce or alter any experimental result, or to write the proofs. All numbers reported in this paper were produced by executing the released code, and every table entry was checked against the committed result logs by the authors. The two propositions were stated by the authors and independently machine-checked in Lean 4, which we regard as a stronger guarantee than review of AI-assisted mathematics. We have reviewed all AI-assisted work and take responsibility for the final content of this paper, including all text, claims and artifacts.

ETHICS STATEMENT

This work involves no human subjects, no personal data, and no operator traces. All data is generated by simulators released with the paper. The application area - radio-access traffic steering - can affect service quality for large user populations, which is the reason we treat safety constraints as hard rather than as penalties, and the reason we report violation counts alongside every utility number in this paper rather than reporting utility alone.

REFERENCES

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *International Conference on Machine Learning*, pp. 22–31, 2017.

Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. Safe reinforcement learning via shielding. In *AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Anastasios N. Angelopoulos and Stephen Bates. Conformal prediction: A gentle introduction. *Foundations and Trends in Machine Learning*, 16(4):494–591, 2023.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15619–15629, 2023.

Mahmoud Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Rina Foygel Barber, Emmanuel J. Candès, Aaditya Ramdas, and Ryan J. Tibshirani. Conformal prediction beyond exchangeability. *The Annals of Statistics*, 51(2):816–845, 2023.

Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations*, 2022.

Felix Berkenkamp, Matteo Turchetta, Angela Schoellig, and Andreas Krause. Safe model-based reinforcement learning with stability guarantees. *Advances in Neural Information Processing Systems*, 30, 2017.

A. Colin Cameron and Pravin K. Trivedi. *Regression Analysis of Count Data*. Cambridge University Press, 2nd edition, 2013.

Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, 2016.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing*, pp. 1724–1734, 2014.

C. K. Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions on Information Theory*, 16(1):41–46, 1970.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Advances in Neural Information Processing Systems*, 31, 2018.

Leonardo de Moura and Sebastian Ullrich. The Lean 4 theorem prover and programming language. In *International Conference on Automated Deduction*, pp. 625–635, 2021.

Bradley Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1): 1–26, 1979.

Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Chapman and Hall/CRC, 1994.

Ran El-Yaniv and Yair Wiener. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11:1605–1641, 2010.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

Javier García and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1):1437–1480, 2015.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. *Advances in Neural Information Processing Systems*, 30, 2017.

Isaac Gibbs and Emmanuel Candès. Adaptive conformal inference under distribution shift. *Advances in Neural Information Processing Systems*, 34:1660–1672, 2021.

Paul Glasserman and David D. Yao. Some guidelines and guarantees for common random numbers. *Management Science*, 38(6):884–908, 1992.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. *Advances in Neural Information Processing Systems*, 31, 2018.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning*, pp. 2555–2565, 2019.

Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *International Conference on Learning Representations*, 2021.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640:647–653, 2025.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*, 2017.

Villy B. Iversen. *Teletraffic Engineering and Network Planning*. Technical University of Denmark, 2010.

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.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. *Advances in Neural Information Processing Systems*, 32, 2019.

Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. MOReL: Model-based offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33: 21810–21823, 2020.

Roger Koenker and Gilbert Bassett. Regression quantiles. *Econometrica*, 46(1):33–50, 1978.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton Earnshaw, Imran Haque, Sara Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. WILDS: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pp. 5637–5664, 2021.

Yann LeCun. A path towards autonomous machine intelligence. *Open Review*, 62(1), 2022.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

David Q. Mayne, James B. Rawlings, Christopher V. Rao, and Pierre O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Do deep generative models know what they don't know? In *International Conference on Learning Representations*, 2019.

Huy Nguyen, Nhat Ho, and Alessandro Rinaldo. Sigmoid gating is more sample efficient than softmax gating in mixture of experts. *Advances in Neural Information Processing Systems*, 37, 2024.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in Neural Information Processing Systems*, 32, 2019.

Natale Patriciello, Sandra Lagen, Biljana Bojovic, and Lorenza Giupponi. An E2E simulator for 5G NR networks. *Simulation Modelling Practice and Theory*, 96:101933, 2019.

Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2nd edition, 2009.

Michele Polese, Leonardo Bonati, Salvatore D'Oro, Stefano Basagni, and Tommaso Melodia. Understanding O-RAN: Architecture, interfaces, algorithms, security, and research challenges. *IEEE Communications Surveys and Tutorials*, 25(2):1376–1411, 2023.

Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence. *Dataset Shift in Machine Learning*. MIT Press, 2009.

James B. Rawlings, David Q. Mayne, and Moritz M. Diehl. *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing, 2nd edition, 2017.

George F. Riley and Thomas R. Henderson. The ns-3 network simulator. In *Modeling and Tools for Network Simulation*, pp. 15–34. Springer, 2010.

Donald B. Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology*, 66(5):688–701, 1974.

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

Uri Shalit, Fredrik D. Johansson, and David Sontag. Estimating individual treatment effect: Generalization bounds and algorithms. In *International Conference on Machine Learning*, pp. 3076–3085, 2017.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017.

The mathlib Community. The Lean mathematical library. In *ACM SIGPLAN International Conference on Certified Programs and Proofs*, pp. 367–381, 2020.

Ryan J. Tibshirani, Rina Foygel Barber, Emmanuel J. Candès, and Aaditya Ramdas. Conformal prediction under covariate shift. *Advances in Neural Information Processing Systems*, 32, 2019.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer, 2005.

Ruofeng Wen, Kari Torkkola, Balakrishnan Narayanaswamy, and Dhruv Madeka. A multi-horizon quantile recurrent forecaster. *arXiv preprint arXiv:1711.11053*, 2017.

Grady Williams, Andrew Aldrich, and Evangelos A. Theodorou. Model predictive path integral control using covariance variable importance sampling. In *IEEE International Conference on Robotics and Automation*, pp. 1714–1721, 2017.

Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. MOPO: Model-based offline policy optimization. *Advances in Neural Information Processing Systems*, 33:14129–14142, 2020.

Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer, and William Fedus. ST-MoE: Designing stable and transferable sparse expert models. *arXiv preprint arXiv:2202.08906*, 2022.

A Extended Related Work

World models and latent prediction. Learning a dynamics model and planning inside it is a long-standing recipe (Ha & Schmidhuber, 2018; Hafner et al., 2019; 2021; 2025). Joint-embedding predictive architectures avoid reconstructing observations and instead predict in a learned latent space (LeCun, 2022; Assran et al., 2023; 2025), usually with an exponential-moving-average target encoder (Grill et al., 2020) and an explicit anti-collapse term (Bardes et al., 2022). Our predictor follows that template, with the difference that its latent must support *calibrated interval* heads, because a shield consumes quantiles rather than point predictions.

Uncertainty in model-based control. Model-based reinforcement learning has long used ensembles or probabilistic dynamics to avoid exploiting model error (Chua et al., 2018; Janner et al., 2019), and the offline variants penalise or truncate rollouts in states the data does not cover (Yu et al., 2020; Kidambi et al., 2020). Our predicted-state check is in that lineage, but it is applied at *decision time in closed loop* rather than as a training-time penalty, and its output is a binary abstention rather than a reward adjustment.

Shielding and constrained control. Shielding synthesises a filter that blocks unsafe actions (Alshiekh et al., 2018); constrained policy optimisation and safe model-based RL enforce constraints during learning (Achiam et al., 2017; Berkenkamp et al., 2017; García & Fernández, 2015). Model predictive control supplies the closed-loop re-planning discipline (Mayne et al., 2000; Rawlings et al., 2017; Williams et al., 2017). The piece we add is that the shield is only trusted while a support test says the quantiles it reads are still meaningful.

Calibration under shift, and abstention. Predictive uncertainty degrades under dataset shift (Ovadia et al., 2019; Quiñonero-Candela et al., 2009; Koh et al., 2021), and simple confidence scores remain a strong baseline for detecting inputs a model should not be trusted on (Hendrycks & Gimpel, 2017) - our self-check plays that role but scores the model against observed dynamics rather than against its own confidence. Conformal prediction gives distribution-free coverage under exchangeability (Vovk et al., 2005; Angelopoulos & Bates, 2023) and has been extended to covariate shift and to bounded departures from exchangeability (Tibshirani et al., 2019; Gibbs & Candès, 2021; Barber et al., 2023). Abstention has an equally old pedigree (Chow, 1970; El-Yaniv & Wiener, 2010; Geifman & El-Yaniv, 2017). Our contribution is not a new estimator in either family but a demonstration that, in a control loop, a coarse support test used to *abstain* recovers more safety than the calibrated intervals themselves can supply once shifted.

Mixtures of experts. Gated mixtures date to Jacobs et al. (1991) and scale through sparse top-k routing (Shazeer et al., 2017; Fedus et al., 2022; Zoph et al., 2022). Softmax gating induces competition among experts; sigmoid gating removes the simplex constraint and has recent sample-efficiency support (Nguyen et al., 2024). We use the distinction as a controlled mechanism probe rather than as a scaling device: with five experts tied to named physical mechanisms, whether gates can fire concurrently is directly interpretable.

Network control. Radio-access control is increasingly posed as a closed-loop learning problem (Polese et al., 2023), and system-level simulators make such studies reproducible (Riley & Henderson, 2010; Patriciello et al., 2019). Our admission model is the classical Erlang-B loss formula (Iversen, 2010). We use the domain because it supplies genuine multi-objective safety constraints, not because our conclusions are specific to it.

B Environment Specifications

B.1 P0-A: analytic digital twin

Per cell i and five-minute bin t, hidden offered load is a diurnal profile modulated by FlashCrowd events, $\lambda_{i,t} = A_i\,(0.75 + 0.25\sin(2\pi t/288 + \phi_i))\,(1 + c_{i,t})$, with $A_i \sim \mathcal{U}(70, 120)$ Erlangs, $\phi_i \sim \mathcal{U}(0, 2\pi)$ and $c_{i,t} \in [0, 3]$ a trapezoidal crowd profile with a Gaussian spatial kernel on the ring adjacency, optionally moving.

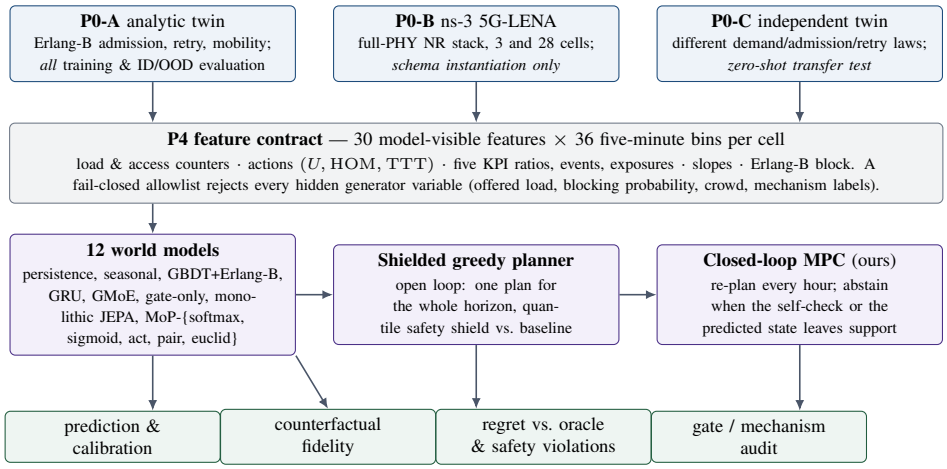


Figure 5: **Benchmark organisation.** Three environments emit the same model-visible feature contract, so a model trained on one can be evaluated on another without any change of interface. All quantitative comparisons in this paper are trained on P0-A; P0-C provides a zero-shot environment-shift test. P0-B instantiates the contract on a running ns-3 5G-LENA NR deployment and is used to check that the interface survives at system scale — it does not contribute training data or model comparisons (Section 7).

Mobility moves load between neighbours according to a logistic of the neighbour load difference, biased by the handover margin and time-to-trigger: $\mathrm{ho}_{i,t}^{\mathrm{out}} = \kappa\,\lambda_{i,t}\,\sigma\big(2\Delta_{i,t}/40 + 2.5\,\frac{8-\mathrm{HOM}}{8} - 1.5\,\frac{\mathrm{TTT}-40}{600} - 0.8\big)$, capped at half the offered load and redistributed equally over neighbours. Admission uses the exact Erlang-B recursion (Iversen, 2010) on the effective load plus retries, with rejected load fed back over the next two bins at fractions ρ and 0.3ρ.

The five indicator rates are affine in blocking probability, a smooth saturation of load over capacity, retry intensity, handover stress and the crowd multiplier, with additive Gaussian noise, clipped to physical ranges. Events are $\mathrm{round}(\text{exposure} \times \text{rate})$ with KPI-specific exposures, so the model always sees integer counts with a matching denominator.

B.2 P0-C: INDEPENDENT TWIN

P0-C is written against the same interface but differs in every mechanism: demand is a weekly harmonic plus a diurnal term multiplied by an exponentiated AR(1) process rather than a single sine with Gaussian noise; blocking uses a Gaussian approximation to Erlang-B rather than the exact recursion; retries decay geometrically over three bins rather than two fixed fractions; mobility uses a hysteretic UE-flow fraction rather than a load-difference spillover; the spatial crowd kernel is Cauchy rather than Gaussian; and indicator noise is multiplicative lognormal rather than additive Gaussian, with different coefficients throughout. Its random-number stream is derived differently, so no exogenous draw is shared with P0-A.

B.3 SCENARIO DISTRIBUTIONS

Table 3 lists the parameters and Figure 6 shows them to scale.

C UTILITY, SAFETY AND REGRET

Let $r_{i,t,j}$ be indicator j for cell i at bin t and $d_j \in \{+1, -1\}$ its direction (-1 for drop rate). For a rollout o against baseline b,

$$\mathcal{U}(o,b) \;=\; w_{\mathrm{conn}}\frac{\overline{\mathrm{expo}}_0^{\,o} - \overline{\mathrm{expo}}_0^{\,b}}{100} \;+\; \sum_j w_j\, d_j\big(\bar{r}_j^{\,o} - \bar{r}_j^{\,b}\big), \tag{4}$$

Table 3: In-distribution and shifted FlashCrowd parameters. The shifted distribution is disjoint from the training distribution in peak amplitude, onset speed, crowd count, movement speed and radius.

Parameter	In-distribution	Shifted (FlashCrowd OOD)
peak amplitude	$\mathcal{U}(0.4, 1.4)$	$\mathcal{U}(1.6, 2.4)$
rise time (fraction of horizon)	$\mathcal{U}(0.02, 0.08)$	$\mathcal{U}(0.005, 0.015)$
concurrent crowds	1	2
movement speed (cells / 100 bins)	$\mathcal{U}(0.1, 0.8)$	$\mathcal{U}(1.2, 2.0)$
spatial radius (cells)	$\mathcal{U}(0.5, 2.0)$	$\mathcal{U}(2.5, 4.0)$
retry fraction ρ	$\mathcal{U}(0.2, 0.5)$	$\mathcal{U}(0.5, 0.7)$
horizon (bins)	1152 train / 288 eval	288
cells	7	7

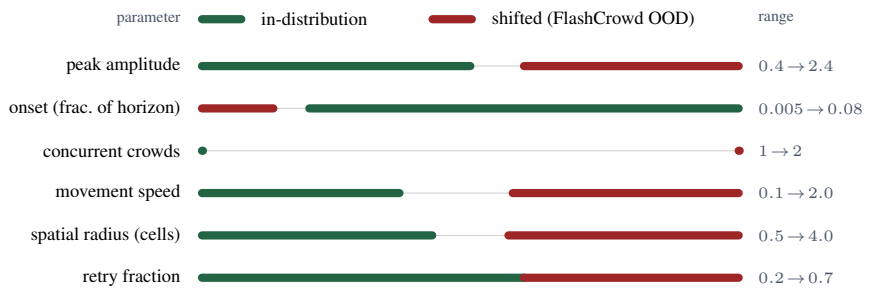


Figure 6: **The shifted scenario distribution is disjoint from the training distribution on five of six parameters.** Each row is normalised to the union of the two ranges, so only the relative position matters; the absolute range is printed at the right. Retry fraction is the single parameter whose ranges touch. This is a deliberate, parameterised shift rather than a natural one, which is a limitation: it makes the shift measurable but also makes it the kind of shift a support test is most likely to catch.

with weights $w_{\text{conn}} = 1.0$, $w_{\text{ssr}} = w_{\text{rasr}} = 0.5$, $w_{\text{cdr}} = 1.5$, $w_{\text{pscell-inter}} = 1.0$, $w_{\text{pscell-intra}} = 0.5$, and bars denoting means over cells and the planning horizon.

Safety is evaluated on $\mathcal{S} = \{\text{cdr}, \text{pscell-inter}\}$ using worst-case-over-time values, with per-indicator tolerances $\varepsilon_{\text{cdr}} = 0.004$ and $\varepsilon_{\text{pscell-inter}} = 0.02$:

$$v_j(o, b) \;=\; d_j\Big(\max_t r_j^{\,o}\Big|_{\text{worst}} - \max_t r_j^{\,b}\Big|_{\text{worst}}\Big) \;-\; \varepsilon_j, \qquad \text{unsafe} \iff \max_{j \in \mathcal{S}} v_j > 0. \tag{5}$$

For higher-is-better indicators the $\max_t$ is a $\min_t$. Learned planners substitute conservative quantiles for $r^{\,o}$: the q_{95} of the drop rate and the q_{05} of inter-cell handover success. The planner additionally applies a search-time margin of 0.004, so a candidate is rejected whenever $v_j + 0.004 > 0$.

Regret is $\mathcal{U}(\text{oracle plan}) - \mathcal{U}(\text{model plan})$ with both plans *realised through the true simulator*; the learned model is used only to choose the plan, never to score it.

D Feature Contract

Table 4 lists the model-visible columns and the generator variables the loader refuses. The allowlist is enforced at load time and unit-tested (Section R), so a model cannot read a hidden mechanism label even by accident.

E Model Architecture and Expert Roles

Figure 7 gives the architecture. Each expert reads a fixed subset of features averaged over a fixed look-back window, which is what makes the gate audit interpretable: an active gate names a mechanism; Table 5 lists the assignments and Figure 8 draws them as a grid.

Table 4: The 30 model-visible features per cell per bin. Everything else in the generator is rejected by a fail-closed allowlist that is unit-tested.

Index	Feature	Note
0-1	`hour`, `dow`	calendar context
2-6	`n_sr`, `n_bsr`, `connected`, `attempts`, `rejects`	load / access counters
7-9	`u`, `hom`, `ttt`	action in the current bin
10-14	`ratio_*`	five indicator ratios
15-19	`event_*`	five indicator numerators
20-24	`expo_*`	five indicator denominators
25-26	`slope_n_sr`, `slope_n_bsr`	five-bin slopes
27-29	`a_post`, `p_block`, `binding`	Erlang-B block from visible counters
Rejected (hidden): `offered_load`, `load_eff`, `retry_in`, `block_prob`, `crowd`, `ho_out`, `ho_in`, `kpi_rates`, `scenario_id`, `seed`, `mech_*`, `oracle_*`		

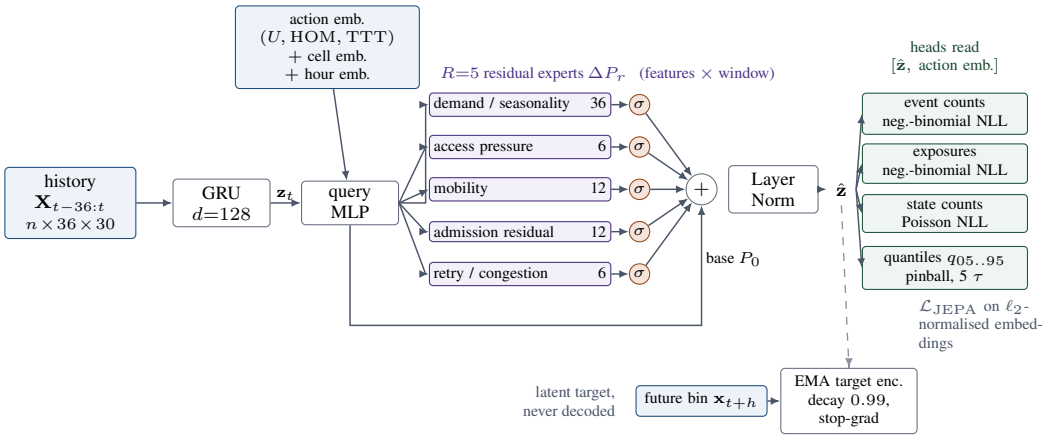


Figure 7: **Sigmoid MoP-JEPA.** A GRU summarises each cell's recent history; a query network fuses that summary with the candidate action, the cell identity and the hour of day. Five residual experts, each restricted to a different subset of features and a different look-back window, are combined through *independent* sigmoid gates rather than a softmax simplex, so any number of mechanisms may be active at once — including none, and including several. The mixture is added to a shared base predictor and normalised, and the resulting latent drives four action-conditioned heads. A separate exponential-moving-average encoder supplies the joint-embedding prediction target. Ablations remove the base, swap the gate for a softmax, or replace the inner-product gate score with a Euclidean one (Section 4).

F Losses and Weights

The total objective is

$$\begin{aligned}\mathcal{L} = w_{\text{jepa}}\mathcal{L}_{\text{jepa}} + w_{\text{var}}\mathcal{L}_{\text{var}} + w_{\text{count}}\mathcal{L}_{\text{count}} + w_{\text{state}}\mathcal{L}_{\text{state}} \\ + w_q\mathcal{L}_{\text{pinball}} + w_{\text{spec}}\mathcal{L}_{\text{spec}}\ (+ w_a\mathcal{L}_{\text{act}} + w_p\mathcal{L}_{\text{pair}}),\end{aligned} \tag{6}$$

with $w_{\text{jepa}} = 1.0$, $w_{\text{count}} = 1.0$, $w_{\text{state}} = 0.3$, $w_q = 0.5$, $w_a = 0.3$, $w_p = 1.0$, $w_{\text{spec}} = 0.1$, $w_{\text{var}} = 0.05$. The last two terms are present only in the corresponding ablations.

The composition of Equation (1) is closed by layer normalisation (Ba et al., 2016), and the interval heads follow the multi-horizon quantile-forecasting recipe (Wen et al., 2017; Salinas et al., 2020) rather than a parametric predictive distribution, so the shield reads the quantiles it needs directly. $\mathcal{L}_{\text{jepa}}$ is the mean squared error between ℓ_2-normalised $\hat{\mathbf{z}}$ and the stop-gradient EMA target. $\mathcal{L}_{\text{var}}$ is a hinge on the standard deviation of $\hat{\mathbf{z}}$ projected onto $\max(8, d/2)$ fixed random directions, $\mathbb{E}[\max(0, 1-\text{std})]$, which penalises collapse along a random subspace rather than the ambient axes. $\mathcal{L}_{\text{spec}}$ combines an activity prior - $\max(0, \bar{g} - 0.5)$ for sigmoid gates, negative entropy for softmax - with the mean absolute off-diagonal cosine similarity between expert outputs, which discourages

Table 5: Expert receptive fields. Windows are in five-minute bins.

Expert	Features	Window
demand / seasonality	`n_sr`, `n_bsr`, `connected`	36
access pressure	`n_sr`, `n_bsr`, `connected`, `attempts`, `rejects`	6
mobility	`u`, `hom`, `ttt`, `ratio_pscell_inter`, `ratio_pscell_intra`	12
admission residual	`u`, `a_post`, `p_block`, `binding`, `rejects`	12
retry / congestion	`rejects`, `ratio_cdr`, `ratio_rasr`	6



Figure 8: **Expert receptive fields.** Each of the five residual experts reads a different slice of the feature contract over a different horizon. This is what makes the gate audit of Figure 10 interpretable: an active gate names a mechanism rather than an anonymous capacity unit. Feature abbreviations follow Table 4; `ps-er` and `ps-ra` are the inter- and intra-cell secondary-cell ratios.

redundant experts. $\mathcal{L}_{\text{pair}}$ aligns predicted and true distances between paired counterfactual rollouts, $\big|\, \|\hat{y}_a - \hat{y}_b\|_1 - \|y_a - y_b\|_1 \,\big|$.

G HYPERPARAMETERS

Table 6 lists every training and planning hyperparameter. All neural models share these values; the only differences between models are the structural switches of Table 7.

Table 6: Training and planning hyperparameters, identical across all neural models unless the model definition overrides them. The optimiser is AdamW (Loshchilov & Hutter, 2019) with a cosine schedule (Loshchilov & Hutter, 2017).

Parameter	Value	Parameter	Value
latent width d	128	optimiser	AdamW
action embedding d_a	32	learning rate	10^{-3}
experts R	5	schedule	2-epoch warmup + cosine
gate temperature τ	0.5	weight decay	10^{-5}
mixture scale α	1.0	gradient clip	1.0
gate bias init	-1.0	epochs	15 (full), 8 (smoke)
EMA decay	0.99	batch size	64
history L	36 bins	input normalisation	per-feature z-score
horizon H	24 bins	window stride	6 (train) / 12 (val)
quantiles τ_q	$\{0.05, 0.10, 0.50, 0.90, 0.95\}$		
decision interval	12 bins	planner steps K	4 (open loop), 1 (closed loop)
minimum gain	0.001	search safety margin	0.004
open-loop action grid	HOM $\in \{0, 8\} \times$ TTT $\in \{40, 320\} \times U \in \{100, 140\}$		
closed-loop action grid	HOM $\in \{0, 8\} \times$ TTT $= 160 \times U \in \{100, 140\}$		
baseline schedule A_0	$U = 120$, HOM $= 4$, TTT $= 160$ for all cells and bins		

H Model Zoo

Table 7 enumerates the twelve models. Figures abbreviate the nine neural models as GRU, GMoE, gate-only, mono-JEPA, MoP-sm, MoP-sg, +act, +pair and +euclid, in the order of the table; MPC-sm and MPC-sg denote the closed-loop controllers of Section 4.3 built on MoP-sm and MoP-sg. The nine neural models differ only in four switches - whether the joint-embedding objective is on, how many experts there are, whether gates are softmax or sigmoid, and whether a shared base predictor carries the prediction - so each comparison in Section 6 isolates one design choice.

Table 7: The twelve models. "Base" is the shared predictor P_0; "residual" means experts add to it rather than replace it.

Model	JEPA	R	gate	base	score	Role
Persistence	-	-	-	-	-	last-bin carry-forward
Seasonal	-	-	-	-	-	same-hour-yesterday
GBDT + Erlang-B	-	-	-	-	-	structured tabular (Chen & Guestrin, 2016)
GRU (monolithic)	✗	0	-	✓	-	sequence baseline
Graph MoE (softmax)	✗	5	softmax	✗	inner	softmax MoE baseline
Gate-only sigmoid MoE	✗	5	sigmoid	✗	inner	sigmoid MoE, no base
Monolithic JEPA	✓	0	-	✓	-	latent-predictive baseline
MoP-JEPA (softmax)	✓	5	softmax	✓	inner	softmax ablation
MoP-JEPA (sigmoid)	✓	5	sigmoid	✓	inner	**main model**
+ action decoder	✓	5	sigmoid	✓	inner	$+\mathcal{L}_{\text{act}}$
+ paired-CF loss	✓	5	sigmoid	✓	inner	$+\mathcal{L}_{\text{pair}}$
+ Euclidean gate	✓	5	sigmoid	✓	euclid	RBF gate score

I Seeds and Evaluation Protocol

Training scenarios use seeds $1000 + j$ for $j < 12$, validation $1100 + j$ for $j < 3$, and in-distribution evaluation $1200 + j$ for $j < 6$; for training seed offset $s \in \{0, 1, 2\}$ the base is $1000 + 10^4 s$. Shifted evaluation uses seeds $2100 + j$ for $j < 30$. All bootstrap intervals resample 5000 times, using the percentile method (Efron & Tibshirani, 1994). Abstention thresholds are the 95th percentile of the self-check statistic over 120 randomly chosen in-distribution training windows, and the support bounds are the 99th percentile of the training drop rate (0.2309) and the 1st percentile of the training setup success (0.3973), each with 0.01 slack. All thresholds are fixed before any shifted scenario is evaluated.

J Counterfactual Fidelity

These quantities are potential-outcome comparisons (Rubin, 1974; Pearl, 2009): because the simulator realises both arms of an intervention under identical exogenous randomness, the treatment effect is observed rather than estimated, which removes the identification burden carried by counterfactual-inference methods (Shalit et al., 2017). Table 8 reports the results. Counterfactual fidelity is measured on paired rollouts sharing exogenous randomness. *PTD* is the mean absolute distance between a predicted trajectory and its true counterpart; *CSE* is the error in the predicted *distance between* two action arms, which measures whether the model places counterfactuals correctly relative to each other even when both are biased; *AS* is a non-saturating action-separation score; *TE-err* is the mean absolute error of predicted treatment effects against the baseline arm.

K Per-Seed Results

Table 9 gives the individual training runs behind the three-seed means of Table 1, and Figure 9 plots the same data. Seed-to-seed variation in control is large - the monolithic GRU ranges from $6/6$ to $3/6$ unsafe runs and its regret varies by almost an order of magnitude - which is precisely why we base the flagship claim on the thirty-scenario shifted evaluation rather than on this table.

Table 8: Counterfactual fidelity, three-seed means. All nine neural models are within noise of one another on every column, which is the point: counterfactual fidelity does not separate them either, yet their control safety differs by more than $4\times$ (Table 1).

Model	PTD ↓	CSE ↓	AS	TE-err ↓
GRU (monolithic)	0.031 ±0.003	0.004	2.24 ±0.12	0.0113
Graph MoE (softmax)	0.032 ±0.003	0.003	2.19 ±0.10	0.0114
Gate-only sigmoid MoE	0.033 ±0.003	0.004	2.18 ±0.12	0.0115
Monolithic JEPA	0.032 ±0.001	0.004	2.06 ±0.08	0.0122
MoP-JEPA (softmax)	0.032 ±0.003	0.003	2.21 ±0.16	0.0112
MoP-JEPA (sigmoid)	0.032 ±0.003	0.004	2.16 ±0.17	0.0115
+ action decoder	**0.029** ±0.003	0.005	1.98 ±0.25	**0.0106**
+ paired-CF loss	0.033 ±0.003	0.004	2.10 ±0.10	0.0120
+ Euclidean gate	0.033 ±0.002	0.004	2.10 ±0.08	0.0114

Table 9: Per-seed in-distribution results. Each cell is regret (unsafe runs of six). Forecasting columns are ratio MAE / coverage / rare-event recall.

	Control: regret (unsafe/6)			**Forecast: MAE / cov. / recall**		
Model	seed 0	seed 1	seed 2	seed 0	seed 1	seed 2
GRU (monolithic)	−.0342 (6)	−.0163 (3)	−.0038 (4)	.0215/.895/.72	.0252/.911/.73	.0234/.884/.78
Graph MoE (softmax)	−.0123 (4)	−.0188 (3)	−.0151 (3)	.0185/.887/.71	.0208/.893/.71	.0194/.891/.77
Gate-only sigmoid MoE	−.0216 (3)	−.0142 (2)	−.0107 (3)	.0192/.886/.72	.0204/.894/.71	.0204/.887/.79
Monolithic JEPA	+.0084 (3)	−.0112 (2)	−.0373 (5)	.0247/.899/.62	.0219/.895/.70	.0218/.886/.76
MoP-JEPA (softmax)	+.0017 (1)	−.0181 (1)	−.0144 (2)	.0187/.888/.73	.0196/.898/.75	.0203/.893/.73
MoP-JEPA (sigmoid)	−.0158 (3)	+.0026 (0)	−.0002 (0)	.0185/.891/.69	.0219/.890/.72	.0200/.890/.75
+ action dec.	+.0077 (0)	+.0023 (0)	+.0039 (0)	.0332/.913/.15	.0356/.911/.16	.0366/.904/.17
+ paired-CF	−.0038 (0)	−.0279 (3)	−.0162 (2)	.0201/.895/.65	.0210/.894/.71	.0205/.895/.73
+ Euclidean	−.0148 (3)	−.0117 (3)	−.0139 (4)	.0198/.893/.68	.0197/.895/.77	.0212/.903/.74

L Complete Metric Table

Table 1 reports the two forecasting columns that order the models most informatively. For completeness, Table 10 gives every remaining quantity the benchmark records: squared-error and interval-width companions to the ratio MAE, the raw count errors the negative-binomial heads are trained on, the realised utilities behind each regret, and the wall-clock cost of planning.

Table 10: All remaining recorded metrics, three-seed means. *RMSE* and *width* accompany the ratio MAE and q_{05}-q_{95} coverage of Table 1; *event* and *expo. MAE* are the raw count errors. *Utility* columns are the realised values whose difference is the regret; the oracle utility is identical across rows because every planner is scored against the same oracle on the same scenarios. *Plan* columns are wall-clock seconds for one open-loop plan on one scenario.

	Interval		**Count MAE**		**Utility**		**Plan (s)**		
Model	RMSE ↓	width	event	expo.	oracle	model	oracle	model	speed-up
GRU (monolithic)	0.0372	0.101	9.98	20.70	0.0047	0.0228	31.4	0.35	89×
Graph MoE (softmax)	0.0312	0.081	13.19	9.73	0.0047	0.0201	31.1	0.91	34×
Gate-only sigmoid MoE	0.0319	0.081	9.91	9.02	0.0047	0.0202	31.1	0.87	36×
Monolithic JEPA	0.0377	0.095	4.85	16.70	0.0047	0.0180	31.2	0.41	76×
MoP-JEPA (softmax)	**0.0312**	**0.080**	9.94	12.62	0.0047	0.0149	31.3	0.91	35×
MoP-JEPA (sigmoid)	0.0314	0.082	17.26	17.27	0.0047	0.0091	31.2	0.75	41×
+ action decoder	0.0677	0.171	12.22	21.83	0.0047	0.0000	31.2	0.26	121×
+ paired-CF loss	0.0324	0.087	**3.98**	24.95	0.0047	0.0206	31.3	0.94	33×
+ Euclidean gate	0.0328	0.089	10.43	9.97	0.0047	0.0181	31.2	0.92	34×

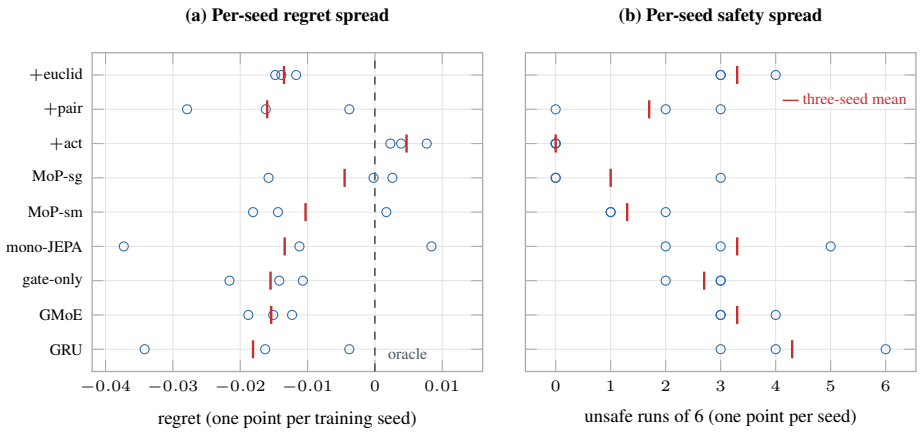


Figure 9: **Seed-to-seed variability of in-distribution control.** Open circles are individual training seeds, red bars the three-seed mean reported in Table 1. The spread is substantial — the monolithic GRU spans 3 to 6 unsafe runs and its regret varies by almost an order of magnitude — which is why the flagship claim of this paper is stated on the thirty-scenario shifted evaluation and why we decline to rank adjacent models here.

Planning cost. The practical argument for a learned world model is that counterfactual search becomes cheap, and the last three columns are the measurement behind it: the identical greedy search costs about 31 s per scenario when each candidate is rolled out through the true simulator, and under a second when it is rolled out through the model. The comparable figures are the 33-41× speed-ups of the models that actually search; the larger numbers belong to models that stop early because they find no admissible action, so they are cheap for the same reason they are safe, and should not be read as an efficiency advantage. Note also that this speed-up is what makes the closed-loop controller of Section 4.3 affordable at all: re-planning hourly with a simulator in the loop would cost roughly 21×31 s per scenario.

Interval width. Coverage alone can be bought with wide intervals, so the width column is the check that it was not. The models with near-nominal coverage in Table 1 carry widths of 0.080-0.101, while the action-decoder ablation reaches its higher coverage of 0.909 with an interval more than twice as wide (0.171) - which is the same fact as its collapsed rare-event recall in Table 9, seen from the other side.

M GATE AUDIT

Table 11: Gate statistics over three seeds, computed on validation windows. A gate counts as active at $g_r > 0.5$. "Redundancy" is the mean absolute off-diagonal correlation between gate activations, so lower means the experts specialise. Models with $R = 0$ are listed for completeness.

Model	active experts (of 5)	all-on rate	all-off rate	redundancy ↓
GRU (monolithic)	0.00 ±0.00	0.000	1.000	-
Monolithic JEPA	0.00 ±0.00	0.000	1.000	-
Graph MoE (softmax)	0.00 ±0.00	0.000	1.000	0.314
Gate-only sigmoid MoE	0.14 ±0.16	0.000	0.855	0.387
MoP-JEPA (softmax)	0.00 ±0.00	0.000	1.000	0.248
MoP-JEPA (sigmoid)	1.54 ±0.68	0.001	0.104	0.230
+ action decoder	2.15 ±0.27	0.083	0.000	0.313
+ paired-CF loss	1.77 ±0.28	0.000	**0.000**	**0.214**
+ Euclidean gate	0.00 ±0.00	0.000	1.000	0.696

Figure 10 plots the audit and Table 11 tabulates it. Two remarks. First, an all-off rate of 1.000 for a softmax gate is not a bug: with five experts on a simplex the maximum achievable gate value is

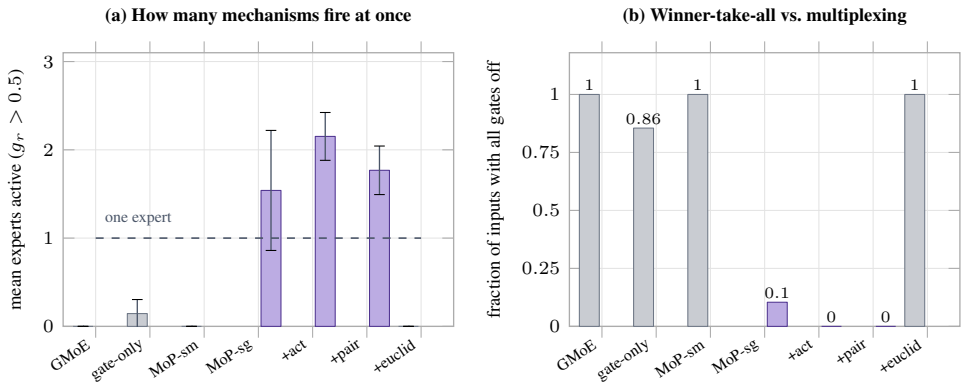


Figure 10: **Gate audit over three training seeds.** A softmax simplex spreads mass so thinly across five experts that no single gate clears the 0.5 activation threshold on any input, so the mixture degenerates to the shared base predictor. Independent sigmoid gates instead keep one to two experts genuinely active, and the paired-counterfactual variant never switches all of them off. The Euclidean-score variant behaves like the softmax models on this diagnostic despite using a sigmoid — its radial-basis score saturates low — which is a useful reminder that "sigmoid gating" is a property of the score as much as of the squashing function. Error bars are one standard deviation across seeds; seed-to-seed spread is large for MoP-sg.

Table 12: **Zero-shot environment shift (train on P0-A, evaluate on P0-C), six scenarios.** The environment model changes while the feature contract stays fixed. Coverage collapses far below the nominal 0.90, and the regret ranking inverts against the safety ranking: the two models with the best regret violate constraints on nearly every scenario, while the model with the worst regret is the only safe one. Open-loop planning throughout.

Model	ratio MAE ↓	pinball ↓	coverage	regret	unsafe /6 ↓
Gate-only sigmoid MoE	0.0588	0.0227	**0.591**	$+0.0112 \pm 0.0124$	1
MoP-JEPA (softmax)	0.0633	0.0253	0.528	-0.0209 ± 0.0140	6
MoP-JEPA (sigmoid)	0.0643	0.0263	0.513	-0.0097 ± 0.0086	5
+ Euclidean gate	**0.0624**	**0.0244**	0.552	$+0.0070 \pm 0.0069$	**0**

bounded below 1 only when the logits are near-uniform, and the learned logits stay close enough to uniform that no gate reaches 0.5. The mixture therefore reduces to the shared base P_0, which is why MoP-sm and Monolithic JEPA have similar forecasting accuracy. Second, MoP-sg's all-off rate has high seed variance (per-seed 0.014, 0.298, 0.000), so the robust statement is that sigmoid variants *can* multiplex and softmax variants never do.

N Zero-Shot Environment Shift

Table 12 gives the full P0-A → P0-C transfer results discussed in Section 6.5: forecast error roughly triples, coverage collapses to 0.51-0.59, and the regret ranking inverts against the safety ranking.

O Six-Seed Shifted Evaluation

Table 13 reports an earlier, smaller sweep on six shifted scenarios; it reproduces the thirty-seed conclusion and additionally reports forecasting metrics under shift, which the thirty-seed run does not. These are the numbers behind the middle bars of Figure 3.

P System-Scale Instantiation with ns-3 5G-LENA

Figure 11 summarises the pipeline and what each stage contributes.

Table 13: Six shifted scenarios, open loop, with forecasting metrics. Coverage falls from ≈ 0.89 in distribution to 0.78-0.81.

Model	ratio MAE ↓	pinball ↓	coverage	regret	unsafe /6
Gate-only sigmoid MoE	0.0535	0.0204	0.783	$+0.0886 \pm 0.0601$	4
MoP-JEPA (softmax)	0.0486	0.0187	0.788	$+0.0468 \pm 0.0360$	1
MoP-JEPA (sigmoid)	0.0510	0.0195	0.781	$+0.0476 \pm 0.0531$	2
+ Euclidean gate	**0.0467**	**0.0180**	**0.810**	$+0.0584 \pm 0.0630$	4
Closed loop with support-aware abstention, same six scenarios					
MoP-JEPA (softmax)	-	-	-	$+0.0087 \pm 0.0062$	**0**
MoP-JEPA (sigmoid)	-	-	-	$+0.0102 \pm 0.0059$	**0**

What was run. We built two ns-3 scenarios against the 5G-LENA NR module (Patriciello et al., 2019): a three-cell scenario simulated for 1800 s producing 9558 trace events, and a 28-cell scenario at 120 kHz numerology (NR numerology 3) with a 3GPP indoor-office channel model, Shannon-model link adaptation, direct-path beamforming, 4×8 gNB antenna arrays, a 20 MHz carrier at 2.12 GHz, and a full EPC and IP stack, run for 300 s at each of five seeds. Traces are aggregated to the five-minute P4 schema by `sim/lena/aggregate.py`, which is covered by three unit tests including one asserting that the no-restart action guard fires on an admission change when the scenario forbids it.

What the data looks like. The 28-cell full-NR episodes give tensors of shape $(28, 1, 5)$ per seed with mean drop rate 0.0239 (range 0.0188-0.0296) and mean setup success 0.9973 (range 0.9503-1.0). A companion 28-cell configuration without the full PHY produces $(28, 2, 5)$ tensors over 600 s per seed.

What this does and does not establish. It establishes that the feature contract can be populated end-to-end from a running system-level NR deployment at the cell count used in the paper's headline scale, and that the adapter and the no-restart semantics survive contact with a real simulator's trace format. It does *not* establish anything about model quality. The counters emitted by these scenarios come from a demand model scheduled as ns-3 events alongside the NR stack rather than from the stack's own radio-level measurements; in particular the two secondary-cell indicators are emitted at fixed success ratios (0.985 and 0.99) and are therefore constant in the resulting episodes. No model in this paper is trained or evaluated on these episodes. Connecting the counters to radio-level traces - RRC connection establishment, handover success callbacks, radio-link failure - is the natural continuation and is not done here.

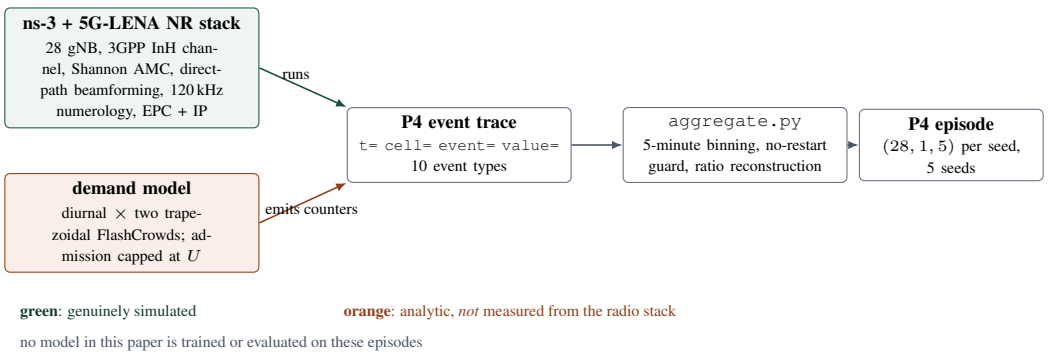


Figure 11: **What the ns-3 instantiation actually shows.** The NR stack is fully configured and genuinely simulated, and the adapter and no-restart semantics are exercised end to end at 28 cells — that is the interface and scale result. The KPI counters, however, are produced by a demand model scheduled alongside the stack rather than read from radio-level measurements, so two of the five indicators are constant by construction. We draw the distinction explicitly here because it bounds what P0-B can be cited for.

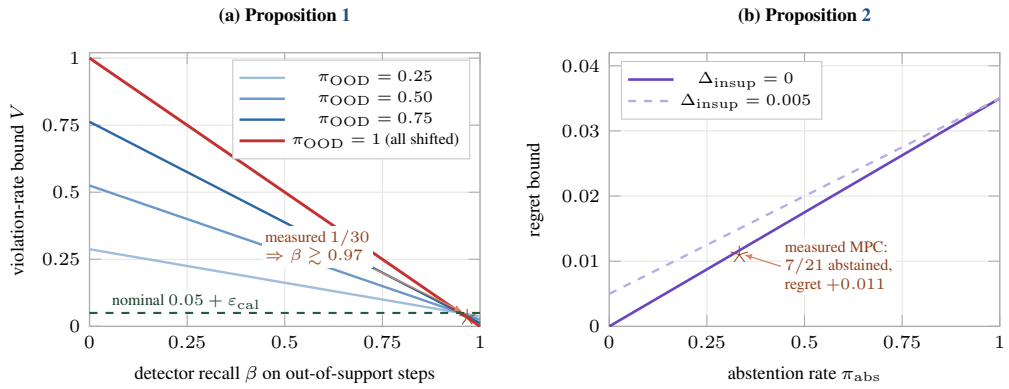


Figure 12: **What the two bounds say, and where the measurements land.** **(a)** The realised violation rate is capped by a convex combination of the calibrated in-support rate and the fraction of shifted steps the detector misses; as recall approaches one the cap collapses onto the nominal level regardless of how much of the workload has shifted. The measured closed-loop rate of one violation in thirty scenarios is consistent with a detector recall above about 0.97 in the worst case where every step is treated as shifted. **(b)** Regret is capped by what abstention forfeits plus what the model loses where it does act. The measured operating point (star) lies on the $\Delta_{\mathrm{insup}} = 0$ line, so the observed regret is accounted for by abstention alone; the plotted ceiling $\bar{U}_{\max} = 0.035$ is illustrative.

Q Machine-Checked Proofs

The two propositions are formalised in Lean 4 with Mathlib (de Moura & Ullrich, 2021; The mathlib Community, 2020). The development contains no `sorry`, no `admit` and no `native_decide`, and ends with an axiom audit: every theorem reports dependence on exactly `[propext, Classical.choice, Quot.sound]`, the three standard Lean axioms.

Table 14 maps each Lean identifier to its statement in the paper.

Table 14: Correspondence between the Lean development and the paper. All identifiers live in the `SafeMPC` namespace.

Lean identifier	Paper statement
`violation_bound`	Proposition 1, Equation (2)
`violation_bound_perfect_recall`	corollary $\beta \to 1$
`price_of_safety`	Proposition 2, Equation (3)
`price_of_safety_exact_on_support`	corollary $\Delta_{\mathrm{insup}} = 0$
`regret_cap_matches_measurement`	regret cap 0.0117 at $\pi_{\mathrm{abs}} = 1/3$
`violation_cap_matches_measurement`	violation cap 0.075 at $\beta = 0.9$, $\pi_{\mathrm{OOD}} = 0.5$

The statement of Proposition 1 in Lean is reproduced below, with the inline comments stripped and the real type $\mathbb{R}$ written `R` for typesetting; the hypothesis `hdecomp` is the law-of-total-probability decomposition over step types, `hVflag` encodes that abstention reproduces the baseline, and the conclusion is Equation (2) verbatim.

```
theorem violation_bound
    (piOOD beta epsCal Vin Vflag Vmiss V : R)
    (hpi0 : 0 <= piOOD) (hpi1 : piOOD <= 1)
    (hb0 : 0 <= beta)   (hb1 : beta <= 1)
    (hVin0 : 0 <= Vin)  (hVin : Vin <= 0.05 + epsCal)
    (hVflag : Vflag = 0)
    (hVmiss0 : 0 <= Vmiss) (hVmiss1 : Vmiss <= 1)
    (hdecomp : V = (1 - piOOD) * Vin
                 + piOOD * (beta * Vflag + (1 - beta) * Vmiss)) :
    V <= (1 - piOOD) * (0.05 + epsCal) + piOOD * (1 - beta)
```

What is verified is the arithmetic step from the decomposition and the calibration and boundedness assumptions to the bound. The decomposition itself is a hypothesis, not a theorem: it is the definition of conditioning on step type. As Section 7 notes, this is not a distribution-free coverage guarantee, and the calibration assumption on in-support steps is exactly the assumption our own measurements show to fail once support is left - which is why the detector, not the calibration, is doing the work.

R UNIT TESTS

Nine tests run in under a second and guard the invariants the benchmark depends on.

Test	Guards
`test_determinism`	identical exogenous draws reproduce identical episodes
`test_no_restart_filter`	admission changes raise when the scenario forbids them
`test_allowlist_fails_closed`	hidden generator variables are rejected by name
`test_paired_counterfactual_common_noise`	paired arms share offered load, differ in a mobility KPI
`test_higher_u_reduces_cdr_when_binding`	the admission mechanism has the right physical sign
`test_ratio_reconstruction`	indicator ratios equal events over exposures
`test_aggregate_ratios_and_shapes`	ns-3 adapter gives right shapes and a known drop ratio
`test_no_restart_guard_on_lena_schedule`	the adapter enforces no-restart semantics
`test_aggregate_full`	adapter counter mapping for attempts, connects, rejects

S COMPUTE

Training one neural model for 15 epochs on 2184 windows takes approximately 150 s. The dominant cost is planning: a single open-loop oracle plan requires roughly 31 s of simulator rollouts (Table 10), and one closed-loop thirty-scenario sweep takes 69 minutes per model. Logged wall-clock for the shifted studies is 1.9 h for the thirty-seed open-loop sweep, 2.3 h for the thirty-seed closed-loop study, 0.5 h for the six-seed closed-loop check and minutes for the cross-environment study; the three-seed full benchmark adds about an hour per seed. The whole suite is therefore roughly 8 GPU-hours. All results were produced on an NVIDIA RTX 3090 and independently reproduced on an Apple M1 with the MPS backend; the two sets of logs are included in the supplement and agree on every qualitative conclusion.

T EXTENDED LIMITATIONS

This section expands Section 7. Each paragraph is the full form of a point stated there in brief.

All quantitative evidence is simulated. No production network data is used. P0-A and P0-C are independently written simulators, which controls for implementation-specific artefacts but not for the possibility that both miss the same feature of real traffic. The conclusion we would defend is about the *relationship* between calibration, shielding and safety, not about absolute drop-rate numbers.

The ns-3 environment is an interface check, not evidence. P0-B instantiates a full 5G-LENA NR stack - 3GPP channel model, Shannon-model link adaptation, direct-path beamforming, 120 kHz numerology, up to 28 gNBs - and we use it to confirm that the feature contract can be populated from a running system-level deployment at scale. However, the P4 counters it emits are produced by a demand model scheduled alongside the NR stack rather than measured from the stack's own radio-level traces; two of the five indicators are consequently constant by construction. No model in this paper is trained or evaluated on P0-B data, and no claim in Section 6 rests on it. Deriving the counters from radio-level measurements is the obvious next step and we have not taken it. We would therefore

describe this work as two-environment evidence with a system-scale interface demonstration, not as three-simulator evidence.

Abstention buys safety with inaction. The controller is safe partly because it declines to act, and it inherits whatever the baseline schedule does. If the baseline were itself drifting into violation, abstention would not protect against it - our safety criterion is baseline-relative by construction. Extending the guarantee to absolute constraints requires a safe fallback controller, which we do not provide.

The propositions are not coverage guarantees. Proposition 1 and Proposition 2 are consequences of conditioning on step type. They assume the quantiles are calibrated *on in-support steps* and take the detector's recall β as given; neither assumption is verified by the theory, and β is not directly measured - Figure 12 reports the recall that would be *consistent with* the observed violation rate, which is a weaker statement. A distribution-free treatment would need the conformal machinery of Barber et al. (2023) or Gibbs & Candès (2021), and our percentile-based support thresholds are heuristics, not conformal predictors.

Statistical resolution. In-distribution control uses six held-out scenarios and three training seeds. That is enough to establish the frontier (Figure 1a) but not to separate adjacent models: 1.0 versus 1.3 unsafe runs is not a distinction we would defend. The shifted evaluation uses thirty scenarios, which is why we put the flagship claim there. Gate activation counts vary substantially across seeds (1.54 ± 0.68 for MoP-JEPA (sigmoid)), so the mechanism claim in Section 6.6 rests on the qualitative all-off contrast, which is stable, rather than on the activation number.

Sigmoid gating is not shown to be more accurate. The flagship paired comparison is a null ($[-0.0010, +0.0015]$), and the softmax variant wins on 18 of 30 seeds. Our claim is confined to mechanism and to in-distribution safety.

The oracle is a greedy oracle. Regret is measured against the same bounded greedy search run on the true simulator, not against a global optimum, and the closed-loop action grid is coarse. Regret should be read as "how much of a specific achievable gain did the learned model forfeit", not as a distance from optimal control.