FIRST VISITS AND REVISITS: LEARNING FROM THE DEPLOYMENT STREAM IN MODEL-BASED CONTROL

Anonymous authors
Paper under double-blind review

ABSTRACT

A controller facing recurring dynamics meets first visits to new dynamics and revisits to dynamics that its stream has shown. We argue that adaptation then runs on two time scales, fast selection among hypotheses the model holds and slow learning of those hypotheses, which compete for the same weights. We propose FSSL (Fast Selection, Slow Learning), a model-based controller whose members keep a frozen copy of their pretrained prediction heads beside a copy that one multiple-choice loss trains between episodes. Each member plans with the head of lowest recent error among both copies. On six streams against eight baselines, FSSL gives the best mean on all six, and slow learning, hard credit and per-head weighting each raise its score significantly. Without the frozen copy, the learning heads lose on first-visit entries and gain on revisits of CartPole and HalfCheetah, whereas with it a frozen head best explains the entered mode for 79 to 100% of the members. Planning-error bounds per phase explain this division of labour. Code and demo are available at `anonymous.4open.science/r/fssl`.

1 INTRODUCTION

A controller deployed under changing dynamics meets two kinds of episodes, whose demands differ even when the dynamics themselves look alike (Xu et al., 2020; Nagabandi et al., 2019b). In a first visit the controller faces dynamics absent from its experience, such as a payload or a terrain that appears for the first time (Iten et al., 2026). In a revisit the dynamics have occurred earlier in the same stream, so the controller already holds transitions that describe them (Alegre et al., 2021). A first visit leaves no time for learning, so it is served by a fast inference over dynamics hypotheses formed before deployment, such as a context or a prediction head (Lee et al., 2020; Seo et al., 2020). A revisit rewards a slower process, learning the hypotheses themselves from the transitions that the stream has already collected (Chua et al., 2018). We therefore read adaptation to recurring dynamics as fast selection and slow learning, two processes whose value each kind of episode measures separately (Li & Kim, 2026). Averaged over a whole stream, the two measures can cancel, hiding a method that wins one kind of episode and loses the other (Figure 1).

Context-identification methods such as CaDM and T-MCL train a dynamics model over source dynamics and identify new dynamics through a context or a prediction head (Lee et al., 2020; Seo et al., 2020). Their weights stay fixed at deployment, so they select fast without learning, and a revisit is served by the same inference as a first visit. GrBAL adapts its model by a gradient step on the last few transitions, but it returns to its meta-learned prior before every planning step (Nagabandi et al., 2019a). Online model learners take the opposite side, fitting an ensemble or a Gaussian process to the stream from scratch and planning with the current fit (Chua et al., 2018; Iten et al., 2026; Xu et al., 2020). Mixture methods such as MOLe keep one model per inferred task and update the responsible models at every time step (Nagabandi et al., 2019b; Xu et al., 2020). Seen through the two time scales, pretrained families commit to fast selection and online families to slow learning.

A controller serving both phases would keep a fast selection among pretrained hypotheses, because that path works before any stream data arrive. It would also learn the hypotheses from the stream, much as residual models correct a fixed prior with data from the target system (Cutler & How, 2015; Saveriano et al., 2017). Both processes would need one credit rule, since training every hypothesis

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

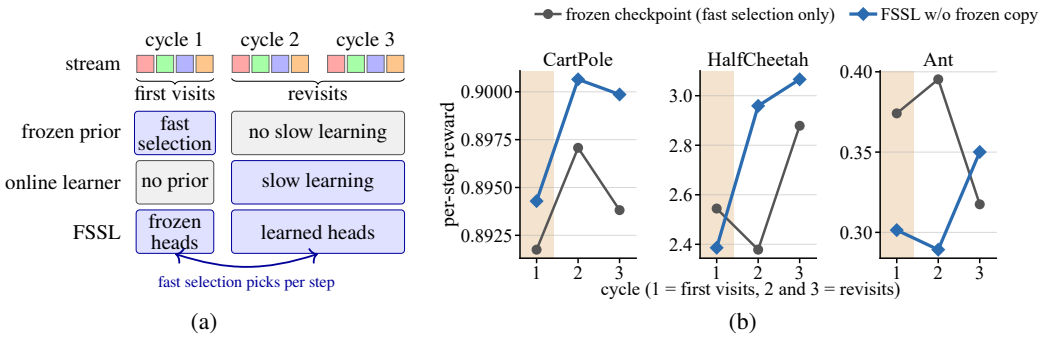


Figure 1: First visits call for fast selection and revisits for slow learning. (a) The first cycle meets every mode for the first time, while later cycles revisit modes whose transitions are already collected. (b) Stream per-step reward by cycle, mean over modes and five seeds, for FSSL w/o frozen copy, which plans with learning heads alone, and for its checkpoint run frozen through the same stream, which selects fast without learning. The learning heads trail the frozen checkpoint in cycle one on HalfCheetah and Ant and end the third cycle above the frozen checkpoint on all three streams.

on every stream transition would pull them toward one average and erase what selection chooses among (Guzman-Rivera et al., 2012; Lee et al., 2016). Continual learning warns, however, that updating shared parameters for the current task can overwrite what those parameters encoded before (Schwarz et al., 2018; Kessler et al., 2023). For a pretrained dynamics model the knowledge at risk is its generalization to unseen dynamics, which is exactly what a first visit relies on. Slow learning could therefore raise revisit performance while lowering first-visit performance, an asymmetry that an average over the stream would hide. What is missing is a model that learns its hypotheses from the stream while keeping the pretrained ones available to fast selection, measured phase by phase.

We propose FSSL (Fast Selection, Slow Learning), a model-based controller whose ensemble members hold their dynamics hypotheses as prediction heads in two copies. A frozen copy keeps the pretrained heads, while a learning copy is updated after every episode by the multiple-choice loss over segments of real transitions that also trains the heads on source dynamics. Each member plans with the head of lowest error on the latest transitions among both copies, so fast selection arbitrates between pretrained and stream-learned hypotheses. On six recurring-dynamics streams against eight baselines, FSSL gives the best mean on all six streams. Under a paired t-test over seeds, removing slow learning, hard credit or per-head weighting lowers the final score on at least one stream. Without the frozen copy, the learning heads gain on the revisits of CartPole and HalfCheetah but lose on the first-visit entries of CartPole, HalfCheetah and Ant, the asymmetry anticipated above. With both copies, a frozen head has the lowest error on the entered mode for most members at first-visit entries, whereas a learning head does so for most members on revisits. Our planning-error bounds explain this division of labour, since buffered data bound the in-support error of the learning heads while a frozen head removes the drift that a first entry meets. We contribute a controller in which fast selection arbitrates between pretrained and learned hypotheses, with a phase-wise diagnosis.

2 Related Work

Model-based control under changing dynamics. Model-based reinforcement learning under changing dynamics differs mainly in whether earlier dynamics return and in what the learner knows before deployment (Iten et al., 2026; Xu et al., 2020; Huang et al., 2021). When dynamics recur, several methods keep one dynamics model per environment and decide online which of the stored models is active (Xu et al., 2020; Nagabandi et al., 2019b; Alegre et al., 2021). GPMM represents each dynamics with a Gaussian process and spawns a new expert when no stored expert explains the incoming transitions (Xu et al., 2020). MBCD detects context changes with a CUSUM statistic on the likelihood of an ensemble and redeploys a stored model-policy pair when a context reappears (Alegre et al., 2021). These methods and online ensembles fit their models to the stream, so their first visits start with little knowledge of the dynamics family (Xu et al., 2020; Chua et al., 2018).

Context identification and meta-learned priors. A second line shares one predictor across environments and encodes the difference between environments in a latent context (Lee et al., 2020; Rakelly et al., 2019; Xie et al., 2021). CaDM encodes recent transitions into a context vector and conditions a single dynamics model on it, training both over a distribution of environments (Lee et al., 2020). These methods fix their weights after training over a family of environments, so a revisit is served by the same inference as a first visit (Lee et al., 2020; Rakelly et al., 2019).

Multiple hypotheses and online adaptation of dynamics models. Multiple-choice learning trains several predictors with an oracle loss that updates the most accurate one, while multiple-model adaptive control acts with the model of smallest recent identification error (Guzman-Rivera et al., 2012; Narendra & Balakrishnan, 1997). T-MCL brings both ideas to dynamics models, crediting trajectory segments to prediction heads that read a learned context and fixing every weight after pretraining (Seo et al., 2020). GrBAL meta-learns a prior that adapts by gradient steps on the last few transitions, and it resets its parameters to that prior before every planning step (Nagabandi et al., 2019a). MOLe keeps a mixture of models under a Chinese-restaurant-process prior and updates every model by one gradient step per time step, weighted by its inferred responsibility (Nagabandi et al., 2019b). Unlike T-MCL and GrBAL, our heads keep learning across episodes, and unlike MOLe they learn between episodes from hard credit on whole segments of the buffer (Seo et al., 2020; Nagabandi et al., 2019a;b). Our model also needs no context encoder, Gaussian variance or backward-prediction loss, because head selection alone identifies the mode (Seo et al., 2020).

Continual learning and new-dynamics detection. Continual learning often fixes by architecture which part of a model is preserved and which part adapts, whereas FSSL keeps a whole frozen copy beside a learning one and lets the latest transitions choose between them (Schwarz et al., 2018; Aljundi et al., 2017). Continual world models such as DRAGO keep memory in data, through selective replay or synthetic rehearsal, often while the dynamics stay fixed and rewards change (Kessler et al., 2023; Sun et al., 2025; Liu et al., 2025). SOMBRL learns a model with optimistic exploration, and R-OMBRL extends it to time-varying dynamics by resetting the data that the model is fit to (Sukhija et al., 2025; Iten et al., 2026). LCPO keeps a policy from forgetting in environments whose dynamics are driven by a changing context (Hamadanian et al., 2025). Deciding whether the current dynamics are new is a change-point problem (Adams & MacKay, 2007; Alegre et al., 2021). Such detectors could tell a first visit from a revisit, a decision that FSSL leaves to head selection over its frozen and learning copies (Adams & MacKay, 2007; Narendra & Balakrishnan, 1997).

3 METHOD

3.1 PRELIMINARIES

Recurring-dynamics stream. A stream is built from K dynamics modes with transition kernels $P_1, \dots, P_K$ over states $s \in \mathcal{S} \subseteq \mathbb{R}^d$ and actions $a \in \mathcal{A}$. It runs for C cycles, and every cycle visits the modes in a fixed order for n consecutive episodes of at most T steps each. The controller observes states, actions and rewards without access to the mode index or to the change points of the stream. We index an episode by its cycle c, its mode k and its position $i \in \{0, \dots, n-1\}$ inside the block of that mode. Its phase follows from the index alone,

$$\phi(c,k,i) = \begin{cases} \text{first-visit entry} & c = 1,\ i = 0,\ k \neq k_{\text{start}}, \\ \text{rest of cycle one} & c = 1 \text{ otherwise}, \\ \text{revisit entry} & c \geq 2,\ i = 0, \\ \text{revisit, settled} & c \geq 2,\ i \geq 1, \end{cases} \tag{1}$$

where k_{start} is the mode that opens the stream. Aggregating equation 1, we call all cycle-one episodes first visits and all episodes of later cycles revisits. The score of an episode is its per-step reward $J = \frac{1}{T_{\text{ep}}} \sum_{t<T_{\text{ep}}} r_t$, where $T_{\text{ep}} \leq T$ is its length. For two controllers run on the same stream and checkpoint, Δ_Φ denotes the mean position-matched difference of J over the episodes of phase Φ. Because every cycle holds the same number of episodes, the stream average splits exactly into the two phases,

$$\Delta_{\text{all}} = \tfrac{1}{C}\, \Delta_{\text{first}} + \left(1 - \tfrac{1}{C}\right) \Delta_{\text{revisit}}, \tag{2}$$

so a positive revisit effect can mask a negative first-visit effect of up to $C - 1$ times its size.

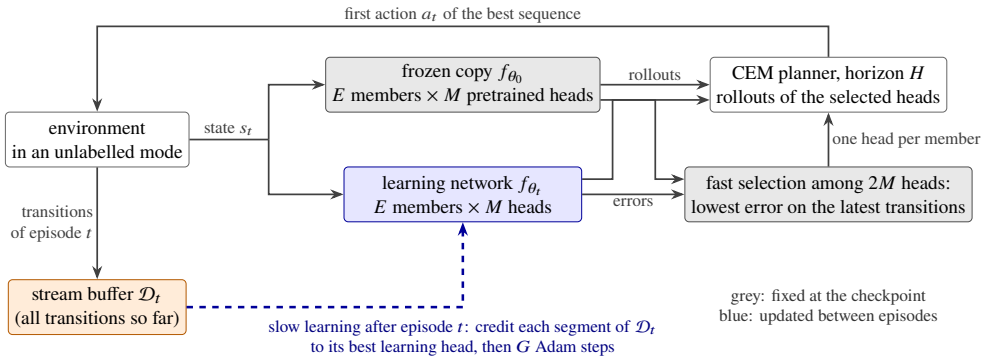


Figure 2: FSSL during one stream episode. Each member holds a frozen copy of its pretrained heads (grey) and a learning copy (blue). Fast selection picks in each member the head of lowest recent error among both copies, and the planner rolls out the selected heads. After the episode, slow learning credits each buffered segment to its best learning head and updates that network.

Ensembles of dynamics models. A model-based controller learns a dynamics model from transitions and chooses its actions by planning through that model (Chua et al., 2018). An ensemble of E networks trained from different initialisations gives several predictions for one input, and their spread signals where the data leave the model uncertain (Lakshminarayanan et al., 2017). Dynamics networks commonly predict the state difference $s_{t+1} - s_t$, standardised by the mean and standard deviation of the training data, which we write $\bar{\Delta}_t$ (Chua et al., 2018).

Multiple-choice learning. Multiple-choice learning trains M predictors $f_1, \ldots, f_M$ on examples (x, y) that may come from several unlabelled modes (Guzman-Rivera et al., 2012). It minimises an oracle loss that credits every example to the predictor that currently explains it best,

$$\mathcal{L}_{\mathrm{MCL}} = \sum_{(x,y)} \min_{m \le M} \ell\big(f_m(x), y\big), \tag{3}$$

so that the gradient of each example reaches the winning predictor alone. This winner-take-all credit lets the predictors specialise to different modes of the data without labels, and it carries over to deep neural networks trained end to end by stochastic gradient descent (Lee et al., 2016).

Receding-horizon planning. Model-predictive control optimises a sequence of H actions under the model, executes its first action and repeats the optimisation at the next step (Camacho & Alba, 2013). The cross-entropy method (CEM) searches this sequence by sampling candidates from a Gaussian refitted to the best ones (De Boer et al., 2005). With an ensemble, each candidate is scored by the mean return of particles propagated through the members (Chua et al., 2018).

3.2 Proposed Method

The framing of Section 1 asks one model to select fast on first visits and to learn slowly on revisits, two processes that FSSL (Fast Selection, Slow Learning) builds from the tools of Section 3.1 (Figure 2). Each ensemble member holds several dynamics hypotheses twice, a frozen copy from pretraining and a learning copy that one multiple-choice loss updates after every episode. At every step, a selection rule picks one hypothesis per member from the latest transitions for the planner.

Several hypotheses per member. This part answers the premise that recurring dynamics fall into a few modes, so a model can hold one hypothesis per group of modes instead of one average. Each of the E members has a shared trunk and M prediction heads, and head m of member e predicts $\bar{\Delta}_t$ deterministically as $f_{\theta,e,m}(s_t, a_t)$, where the trunks and heads form the forward network θ.

One multiple-choice loss from source to stream. This part answers the missing mode labels, because the dynamics stay fixed within an episode and a short run of its transitions therefore carries

evidence for one mode. For a segment τ of L_s consecutive transitions of one episode, head m of member e has the error

$$\ell_{e,m}(\tau;\theta) = \frac{1}{L_s\, d} \sum_{t\in\tau} \left\| f_{\theta,e,m}(s_t, a_t) - \bar{\Delta}_t \right\|_2^2, \tag{4}$$

the squared error averaged over the transitions and the state dimensions. Every segment of a batch $\mathcal{B}$ is credited to the head with the lowest error, which splits the batch into sets $\mathcal{B}_{e,m}$ without gradient through the credit,

$$\mathcal{L}_{\mathrm{S}}(\theta;\mathcal{B}) = \sum_{e=1}^{E} \sum_{m=1}^{M} \frac{1}{\max(|\mathcal{B}_{e,m}|, 1)} \sum_{\tau\in\mathcal{B}_{e,m}} \ell_{e,m}(\tau;\theta), \tag{5}$$

the oracle loss of equation 3 with segments as examples and one mean per head, the per-head weighting. Pretraining on source modes that exclude every stream mode applies this credit once per epoch on a fixed grid of segments, fitting single transitions that carry the credit of their segment, with a weight decay and a warm-up in which every head first fits every transition (Table C.2). T-MCL first combined segment credit with head selection for dynamics models, together with a context encoder, a Gaussian variance and a backward-prediction loss (Seo et al., 2020). Our checkpoint keeps the deterministic multiple-choice core without them, and the next three parts add what it lacks.

Slow learning on the stream. This part answers the revisit half of the framing, because a revisit is the phase in which the stream already holds data of the current mode. The controller keeps a buffer $\mathcal{D}_t$ with every transition of the stream up to the end of episode t, without mode labels and without any other data. After episode t ends, the forward network takes G Adam steps on equation 5 with batches drawn from $\mathcal{D}_t$, starting from θ_{t-1},

$$\theta_t = \mathrm{Adam}_G\big(\theta_{t-1};\ \nabla_\theta\, \mathcal{L}_{\mathrm{S}}(\theta;\mathcal{B}\sim\mathcal{D}_t)\big), \qquad \theta_{\cdot}\big|_{t=0} = \theta_0. \tag{6}$$

Each step draws B segments per member at random positions inside single episodes and recomputes their credit. We use $G = 150$ steps, learning rate 10^{-4} and $B = 16$.

A frozen copy beside the learning heads. This part answers the first-visit half of the framing once slow learning has moved the heads, because a first entry to a mode meets a drift that no term of equation 6 constrains (Lemma 1.2). Each member keeps a frozen copy of its whole pretrained network θ_0 next to the network θ_t that slow learning updates, so it holds $2M$ heads, M from pretraining and M from the stream. On a mode that the stream has not shown, a frozen head predicts as the checkpoint did, whereas on a revisit a learning head fits the buffered data of the mode. Credit and gradients of equation 6 reach the learning heads only, so the copy adds no trained weight.

Fast selection among both. This part lets the latest transitions decide between the two sets, because a selection needs no gradient step and can use the first transitions of a new episode. At step t, each member plans with the head of lowest prediction error among its $2M$ heads on at most nine earlier transitions of the current episode, the selection rule of T-MCL, and with a random head before the second transition. This evidence restarts with every episode, so a switch of the dynamics is answered within a few steps by another head, frozen or learning. Pretrained and stream knowledge thus compete by their error on the current episode, with no rule that fixes the phase in advance.

Receding-horizon planning. The planner runs CEM over 200 candidate sequences of horizon $H = 30$, scores them with a known planning reward $\hat{r}$ and replans at every step. Each member rolls out its selected head over the whole horizon, and a candidate is scored by the mean return over the members. We use $E = 5$ members, $M = 3$ heads, $L_s = 10$ and 20 particles, as in T-MCL.

Complexity. The stream update costs G forward-backward passes per episode over $E \cdot B \cdot L_s$ transitions, a cost independent of the stream length and of the number of modes.

Remark 1 (Cost). The update runs between episodes, so its cost adds to the wall-clock time of the stream without entering the planning loop of any single control step (Section 4.2).

Phase-wise planning guarantees. The results below ask what a planner that rolls out one learning predictor can guarantee in each phase of equation 1. We take the dynamics of mode k as a map $F_k(s,a)$, which holds for our simulated streams up to their reset noise, and write $\hat{f}_\theta(s,a)$ for the de-normalised next state that the selected head predicts. We assume rewards $r(\cdot,a)$ and dynamics $F_k(\cdot,a)$ that are L_r- and L_F-Lipschitz in the state, with the pointwise model error $g_\theta(s,a) = \|\hat{f}_\theta(s,a) - F_k(s,a)\|$. For a start state s and a candidate $\mathbf{a} = (a_0,\dots,a_{H-1})$ in the searched set $\mathcal{C}$, the model rollout starts at $\hat{s}_0 = s$ and follows $\hat{s}_{u+1} = \hat{f}_\theta(\hat{s}_u, a_u)$. We write $R_H(s,\mathbf{a};f) = \sum_{u<H} r(s_u,a_u)$ for the return of the rollout under a map f. The model error along the rollouts of the planner is

$$\varepsilon_k(\theta;s) = \max_{\mathbf{a}\in\mathcal{C}} \max_{u<H} g_\theta(\hat{s}_u,a_u), \qquad \kappa_H = L_r \sum_{u=1}^{H-1}\sum_{j=0}^{u-1} L_F^j. \tag{7}$$

Proposition 1 (Planning gap from state-space model error)**.** *Let the planning reward equal r. If $\hat{\mathbf{a}}$ maximises $R_H(s,\cdot\,;\hat{f}_\theta)$ over $\mathcal{C}$, then $R_H(s,\hat{\mathbf{a}};F_k) \geq \max_{\mathbf{a}\in\mathcal{C}} R_H(s,\mathbf{a};F_k) - 2\kappa_H\,\varepsilon_k(\theta;s)$, and the same holds for the mean return over particles with ε_k taken over the heads they use.*

The two lemmas below relate this error to the data that slow learning sees and to the weight change.

Lemma 1.1 (What the stream constrains)**.** *Let $\{x_i\}_{i\leq N_k}$ be the buffered state-action pairs of mode k, $m_k(\theta) = \frac{1}{N_k}\sum_i g_\theta(x_i)^2$ their mean squared error in raw units, and $S_k(r)$ the set of pairs within distance r of some x_i. (i) If g_θ is L_g-Lipschitz on $S_k(r)$, then $\sup_{S_k(r)} g_\theta \leq \sqrt{N_k\, m_k(\theta)} + L_g r$. (ii) On the first-visit entry of mode k in episode t, $\mathcal{D}_{t-1}$ holds no mode-k transition, so the deployed θ_{t-1} does not depend on F_k.*

Lemma 1.2 (Drift on unseen dynamics)**.** *For all weights θ,θ_0 and every pair (s,a), $|g_\theta(s,a) - g_{\theta_0}(s,a)| \leq \Delta_{\theta,\theta_0}(s,a) = \|\hat{f}_\theta(s,a) - \hat{f}_{\theta_0}(s,a)\|$.*

Remark 2 (The final evaluation measures revisits). Table 1 evaluates frozen weights after the whole stream, when every mode has appeared in C cycles. Part (ii) of Lemma 1.1 then applies to no mode, so Table 1 does not measure the drift Δ_{θ,θ_0} of Lemma 1.2 on first visits to a mode.

Theorem 1 (Phase asymmetry of one learning predictor)**.** *Let the planner of episode t roll out $\hat{f}_{\theta_{t-1}}$ from equation 6 under the assumptions of Proposition 1, and let the particles of each member use the head credited with the buffered segments of the current mode. (i) On any episode t of mode k after its first, if every rollout pair lies in $S_k(r)$, the planning gap is at most $2\kappa_H\big(\sqrt{N_k\, m_k(\theta_{t-1})} + L_g r\big)$, where m_k is the buffered mode-k mean squared error of the credited head. (ii) On a first-visit entry, the planning gap is at most $2\kappa_H \max_{\mathbf{a},u}\big[g_{\theta_0}(\hat{s}_u,a_u) + \Delta_{\theta_{t-1},\theta_0}(\hat{s}_u,a_u)\big]$ on the rollouts of $\hat{f}_{\theta_{t-1}}$.*

Part (i) covers revisits and the later episodes of the first block, whose buffers hold data of the mode, whereas (ii) covers the first-visit entry alone. A member that selects another head falls outside part (i), and Proposition 1 then takes ε_k over every selected head, so the bound tightens once selection picks the credited head. Slow learning can tighten the bound of (i), whereas (ii) adds a drift that no term of the objective observes. The frozen copy of FSSL gives selection a drift-free head.

Corollary 1.1 (A frozen copy removes the drift term)**.** *Let each member select among the heads of θ_{t-1} and of θ_0. On a first-visit entry on which every member selects a head of θ_0, the planning gap is at most $2\kappa_H \max_{\mathbf{a},u} g_{\theta_0}(\hat{s}_u,a_u)$ on the rollouts of $\hat{f}_{\theta_0}$, the bound of (ii) with $\Delta_{\theta_0,\theta_0} = 0$. On a later episode of the mode on which every member selects its credited learning head, (i) holds unchanged.*

4 Experiments

4.1 Experimental Settings

Environments and streams. We use six recurring-dynamics streams, CartPole swing-up, HalfCheetah, Ant, Intersection, SlimHumanoid and Hopper, with one fixed stream per environment. CartPole varies the pole mass and length, and the four MuJoCo streams vary the body mass, with no stream mode in any pretraining source set (Table C.1). Intersection follows the GPMM protocol, where the start and goal entries of a surrounding vehicle define three modes for an ego vehicle with continuous acceleration and steering (Xu et al., 2020). Intersection runs six cycles of three episodes

Table 1: Per-step reward of the frozen final evaluation after the stream, mean ± population SD over five training seeds (higher is better, best in bold). ▼/▲: significantly lower/higher than FSSL (two-sided paired t-test over the five seeds paired by seed index, $p < 0.05$). Protocol notes are in Appendix B. †Four seeds, one seed diverged.

Method	CartPole	HalfCheetah	Ant	Intersection	SlimHumanoid	Hopper
FSSL (ours)	$\mathbf{0.901 \pm 0.001}$	$\mathbf{3.478 \pm 0.794}$	$\mathbf{0.342 \pm 0.071}$	$\mathbf{0.998 \pm 0.070}$	$\mathbf{2.715 \pm 2.065}$	$\mathbf{2.496 \pm 0.121}$
Context identification						
T-MCL	$0.861 \pm 0.021^{\blacktriangledown}$	$2.227 \pm 0.601^{\blacktriangledown}$	0.225 ± 0.061	$0.620 \pm 0.177^{\blacktriangledown}$	-0.247 ± 2.425	2.445 ± 0.052
FAMLE	$0.701 \pm 0.103^{\blacktriangledown}$	$-0.009 \pm 0.127^{\blacktriangledown}$	$0.044 \pm 0.058^{\blacktriangledown}$	$0.418 \pm 0.131^{\blacktriangledown}$	$-2.103 \pm 0.595^{\blacktriangledown}$	$1.686 \pm 0.125^{\blacktriangledown}$
Online model adaptation						
MOLe	$0.375 \pm 0.169^{\blacktriangledown}$	$0.877 \pm 0.283^{\blacktriangledown}$	0.248 ± 0.062	0.885 ± 0.055	$-1.329 \pm 1.092^{\blacktriangledown}$	$2.129 \pm 0.215^{\dagger}$
WorldAgen	$0.669 \pm 0.125^{\blacktriangledown}$	$0.365 \pm 0.079^{\blacktriangledown}$	0.224 ± 0.069	$0.376 \pm 0.210^{\blacktriangledown}$	$-2.499 \pm 0.703^{\blacktriangledown}$	$2.332 \pm 0.075^{\blacktriangledown}$
GrBAL	$0.207 \pm 0.094^{\blacktriangledown}$	$0.657 \pm 0.179^{\blacktriangledown}$	0.252 ± 0.035	$0.568 \pm 0.167^{\blacktriangledown}$	$-0.192 \pm 0.573^{\blacktriangledown}$	$1.449 \pm 0.521^{\blacktriangledown}$
Continual and time-varying RL						
SOMBRL	$0.556 \pm 0.058^{\blacktriangledown}$	$0.158 \pm 0.077^{\blacktriangledown}$	$0.119 \pm 0.066^{\blacktriangledown}$	$0.319 \pm 0.007^{\blacktriangledown}$	$1.335 \pm 2.120^{\blacktriangledown}$	1.352 ± 0.974
LCPO	$0.026 \pm 0.006^{\blacktriangledown}$	$-0.380 \pm 0.069^{\blacktriangledown}$	$0.070 \pm 0.050^{\blacktriangledown}$	$0.278 \pm 0.034^{\blacktriangledown}$	$-2.800 \pm 0.488^{\blacktriangledown}$	$0.805 \pm 0.151^{\blacktriangledown}$
R-OMBRL	$0.025 \pm 0.005^{\blacktriangledown}$	$0.057 \pm 0.075^{\blacktriangledown}$	$0.040 \pm 0.046^{\blacktriangledown}$	$0.366 \pm 0.043^{\blacktriangledown}$	0.357 ± 3.791	$2.133 \pm 0.108^{\blacktriangledown}$

of at most 41 steps per mode, while the other streams run three cycles of three 200-step episodes per mode. No method receives mode labels or change points, while every model-based planner scores its candidates with the same declared closed-form planning reward for each stream.

Baselines. The baselines fall into the three groups of Table 1, and none of them receives task labels at deployment. T-MCL (Seo et al., 2020) and FAMLE (Kaushik et al., 2020) pretrain on the declared source sets and identify the dynamics from recent transitions. MOLe (Nagabandi et al., 2019b), reimplemented on the official GrBAL prior (Nagabandi et al., 2019a), adapts a mixture of models online, GrBAL (Nagabandi et al., 2019a) adapts that prior by a gradient step on recent transitions, and WorldAgen (Wan et al., 2026) adapts by test-time training after pretraining on demonstrations. SOMBRL (Sukhija et al., 2025) and R-OMBRL (Iten et al., 2026) learn a model ensemble and a policy from the stream alone, and LCPO (Hamadanian et al., 2025) learns a policy with one constrained policy update per episode (protocol notes for every row in Appendix B).

Protocol and metrics. After the stream, every method is evaluated with frozen weights on every mode for three episodes, with environment seed $190000 + 1000 \cdot \text{mode} + \text{episode}$. We report the mean per-step reward over five training seeds with the population standard deviation over seeds. FSSL pretrains its own checkpoints for seeds 0 to 4 on the same source sets and budget as the context-identification rows. Phase analyses on CartPole, HalfCheetah and Ant pair each model with its frozen checkpoint, and FSSL with FSSL w/o frozen copy, matching episodes by cycle, mode and position. We report 90% bootstrap intervals that resample the five training seeds as clusters, together with the number of seeds whose paired difference shares the sign of the mean. Development runs use separate checkpoints with seeds 124 to 138, disjoint from the Table 1 seeds 0 to 4. Appendix C gives the full settings of every model and the software environment of the runs.

4.2 Experimental Results

Main comparison. Table 1 compares FSSL with the eight baselines under the frozen final evaluation, which by Remark 2 measures the revisit regime. FSSL gives the highest mean on all six streams. The analyses below first isolate the learning heads, FSSL w/o frozen copy, against the frozen checkpoint, and then ask what the frozen copy adds to them (Appendix D).

When the learning heads pay: first visits against revisits. Figure 3 compares both models with the checkpoint frozen through the same stream, phase by phase. Without the frozen copy, the first-visit entries lose on all three streams with intervals below zero, from -0.012 $[-0.025, -0.001]$ on CartPole to -0.941 $[-1.367, -0.394]$ on HalfCheetah. The same learning heads gain on revisits, by $+0.384$ $[+0.097, +0.669]$ on HalfCheetah and $+0.005$ $[+0.001, +0.008]$ on CartPole, in four of five seeds each. This matches the asymmetry of Theorem 1, since a first entry meets the drift of Lemma 1.2 while a revisit meets heads already fitted to the buffered data of its mode.

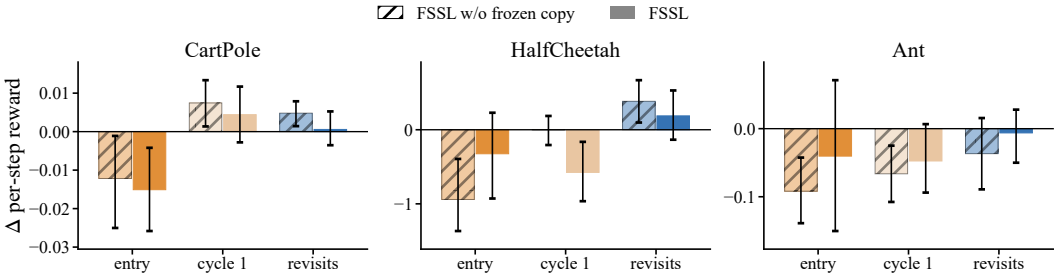


Figure 3: Phase asymmetry and the frozen copy (Theorem 1, Corollary 1.1). Paired per-step difference against the frozen checkpoint for first-visit entries, the rest of cycle one and revisits, for FSSL (solid) and FSSL w/o frozen copy (hatched), with 90% seed-cluster bootstrap intervals.

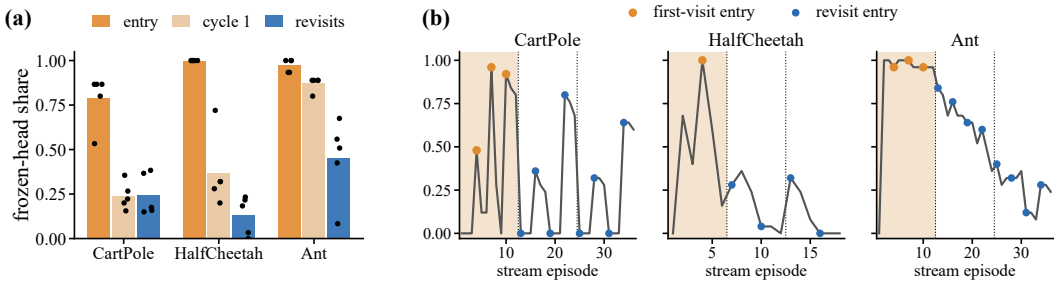


Figure 4: Which copy has the lowest error on the current mode (Corollary 1.1). (a) Share of members whose lowest-error head on the fixed state-action set of the current mode is frozen, by phase, mean over five seeds (bars) and per seed (dots). (b) The same share per stream episode, mean over five seeds, with first-visit and revisit entries marked. Shading marks cycle one, the first visits.

What the frozen copy takes over. Corollary 1.1 removes the drift only when selection picks a frozen head at a first-visit entry, so we measure which copy has the lowest error in each phase on fixed state-action sets of every mode (Appendix D). At first-visit entries a frozen head has the lowest error on the entered mode for 79%, 100% and 97% of the members on CartPole, HalfCheetah and Ant (Figure 4a). On revisits these shares fall to 25%, 13% and 45%, and in all five seeds of every stream the share at entries exceeds the share on revisits. Along the stream the share peaks at first-visit entries and falls on later entries to the same modes, most steadily on Ant, where the learning heads take over as the buffer grows (Figure 4b). With the frozen copy the entry loss shrinks to -0.324 $[-0.928, +0.230]$ on HalfCheetah and -0.040 $[-0.150, +0.071]$ on Ant. Against FSSL w/o frozen copy, FSSL raises the HalfCheetah entry reward by $+0.617$ $[+0.086, +1.085]$ in four of five seeds.

What slow learning, the frozen copy and each component add. Table 2 removes one component at a time and tests every column with the paired t-test of Table 1. Slow learning raises the final score on CartPole by $+0.007$ ($p = 0.004$) and on HalfCheetah by $+0.623$ ($p = 0.037$), while hard credit and the per-head weighting of equation 5 raise CartPole by $+0.012$ ($p = 0.015$) and $+0.003$ ($p = 0.027$). The same two parts raise HalfCheetah by $+0.944$ and $+0.778$ ($p = 0.065$ and 0.070), and the frozen copy raises SlimHumanoid by $+1.327$ ($p = 0.136$). With the 90% seed-cluster intervals over matched episodes, every component raises at least one column in four or five of five seeds (Table D.1). The frozen copy acts on first-visit entries during the stream (Figures 3 and 4), a phase that the final evaluation does not measure (Remark 2). Before any update, where the two copies coincide, removing the Gaussian variance alone raises frozen T-MCL by $+0.040$, $+0.706$ and $+0.131$ on CartPole, HalfCheetah and Ant and brings it within 0.078 of our frozen checkpoint on all three streams (Table 3). The checkpoint of FSSL thus shares the deterministic multiple-choice core of T-MCL, while slow learning and the frozen copy add what the frozen checkpoint lacks.

Table 2: Component ablations of FSSL: final per-step reward of FSSL minus each ablation, mean over five seeds, with the p-value of the two-sided paired t-test over the seeds paired by seed index in parentheses (the test of Table 1). Bold: the component raises the score with $p < 0.05$.

removed component	CartPole	HalfCheetah	Ant	Intersection	SlimHumanoid	Hopper
w/o frozen copy	-0.001 (0.667)	$+0.021$ (0.939)	-0.051 (0.393)	-0.033 (0.381)	$+1.327$ (0.136)	$+0.045$ (0.650)
w/o hard credit	$\mathbf{+0.012}$ (0.015)	$+0.944$ (0.064)	-0.010 (0.832)	-0.039 (0.655)	-0.129 (0.900)	$+0.030$ (0.646)
w/o head selection	$+0.002$ (0.349)	$+0.340$ (0.273)	-0.089 (0.106)	$+0.303$ (0.157)	-0.885 (0.053)	-0.031 (0.618)
w/o per-head weighting	$\mathbf{+0.003}$ (0.027)	$+0.778$ (0.070)	$+0.019$ (0.745)	-0.010 (0.834)	$+0.941$ (0.305)	-0.026 (0.658)
w/o slow learning	$\mathbf{+0.007}$ (0.004)	$\mathbf{+0.623}$ (0.037)	-0.002 (0.952)	$+0.243$ (0.114)	$+1.184$ (0.414)	-0.053 (0.493)

Table 3: Frozen T-MCL checkpoints with one part removed, each pretrained anew for the seeds of Table 1 and run frozen through the stream: final per-step reward of the checkpoint without the part minus frozen T-MCL, and frozen FSSL minus the checkpoint without the part. All five seeds are run, and $k/5$ in parentheses counts the seeds whose difference has the sign of the mean, brackets the 90% seed-cluster interval.

removed part	difference	CartPole	HalfCheetah	Ant
backward loss	it $-$ T-MCL	$+0.018$ (4/5) $[+0.006, +0.031]$	-0.006 (4/5) $[-0.464, +0.630]$	-0.039 (3/5) $[-0.102, +0.021]$
	frozen FSSL $-$ it	$+0.014$ (4/5) $[+0.004, +0.025]$	$+0.790$ (5/5) $[+0.402, +1.201]$	$+0.207$ (5/5) $[+0.161, +0.247]$
Gaussian variance	it $-$ T-MCL	$+0.040$ (5/5) $[+0.022, +0.060]$	$+0.706$ (4/5) $[+0.190, +1.222]$	$+0.131$ (5/5) $[+0.061, +0.212]$
	frozen FSSL $-$ it	-0.008 (5/5) $[-0.011, -0.006]$	$+0.078$ (3/5) $[-0.371, +0.578]$	$+0.037$ (4/5) $[-0.028, +0.104]$
context encoder	it $-$ T-MCL	$+0.015$ (4/5) $[-0.009, +0.038]$	-0.077 (2/5) $[-0.616, +0.425]$	-0.027 (4/5) $[-0.101, +0.048]$
	frozen FSSL $-$ it	$+0.017$ (4/5) $[-0.001, +0.035]$	$+0.861$ (4/5) $[+0.145, +1.416]$	$+0.195$ (5/5) $[+0.135, +0.254]$

Cost of learning on the stream. The update of FSSL costs 5.4 to 10.6 ms per environment step on the 200-step streams, 4.2 to 9.2% of the planning time in the same runs, and 27.0% on Intersection, whose short episodes spread the same update over fewer environment steps (Remark 1).

5 Conclusion

This paper read adaptation to recurring dynamics as fast selection among pretrained hypotheses and slow learning of those hypotheses from the stream. FSSL keeps a frozen copy of its pretrained prediction heads beside a learning copy, trains the learning copy by one multiple-choice loss over segments of real transitions, and plans in each member with the head of lowest recent error among both. It gives the best mean on all six streams, and slow learning, hard credit and per-head weighting each raise its final score significantly on at least one stream. The learning heads alone gain on revisits of CartPole and HalfCheetah but lose on first-visit entries of all three analysed streams, and on fixed state-action sets frozen heads err least at these entries while learning heads do so on revisits. Planning-error bounds per phase explain this division of labour between the copies, since buffered data bound the in-support error of the learning heads while a frozen head removes the drift that a first entry meets.

Under the paired t-test the frozen copy and head selection raise no final score significantly, and on HalfCheetah the frozen copy lowers the rest of cycle one by 0.574 $[0.280, 0.883]$ against FSSL w/o frozen copy. Over the whole stream FSSL stays within the intervals of its frozen checkpoint on CartPole, HalfCheetah and Ant, and its CartPole entry still loses 0.015 per step, so the deployment reward does not yet follow the final score. The streams are low-dimensional and recur over a few fixed modes, and measuring the frozen-head shares on the selections of the planner would test this division more directly. Our bounds rest on Lipschitz, coverage and credited-head conditions that we do not measure on the planner rollouts, while five seeds limit the power of every paired comparison.

AI USE STATEMENT

We used large language models to aid and polish the writing, to retrieve and discover related work, to draft sections of the paper, to assist in proving the mathematical claims, and to assist in research ideation and execution. In execution, the models wrote experiment code, launched training and evaluation jobs, and analysed their logs. The authors checked every AI-assisted output, the code by smoke runs before the full jobs, each comparison by an analysis script fixed by its hash before the results existed, every reported number by the script that computes it from the run logs, the related-work statements against the cited papers, and every proof step against the stated assumptions. The authors take full responsibility for every statement, number and claim of the paper.

REPRODUCIBILITY STATEMENT

The streams, source sets and evaluation protocol are given in Section 4.1, Appendix B and Table C.1, the objectives and the hyperparameters of FSSL in Appendix C and Table C.2, the proofs in Appendix A, and the ablation protocols in Appendix D, while the anonymous repository linked in the abstract contains the code of FSSL and an offline demo built from the stream logs of the paper.

REFERENCES

Ryan Prescott Adams and David J. C. MacKay. Bayesian online changepoint detection. *arXiv preprint arXiv:0710.3742*, 2007.

Lucas N. Alegre, Ana L. C. Bazzan, and Bruno C. da Silva. Minimum-delay adaptation in non-stationary reinforcement learning via online high-confidence change-point detection. In *Proceedings of the 20th International Conference on Autonomous Agents and Multiagent Systems*, pp. 97–105, 2021. arXiv:2105.09452.

Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. Expert gate: Lifelong learning with a network of experts. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. arXiv:1611.06194.

E. F. Camacho and C. B. Alba. *Model Predictive Control*. Springer Science & Business Media, 2013.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems*, volume 31, 2018. arXiv:1805.12114.

Mark Cutler and Jonathan P. How. Efficient reinforcement learning for robots using informative simulated priors. In *IEEE International Conference on Robotics and Automation*, pp. 2605–2612, 2015.

Pieter-Tjerk De Boer, Dirk P. Kroese, Shie Mannor, and Reuven Y. Rubinstein. A tutorial on the cross-entropy method. *Annals of Operations Research*, 2005.

Abner Guzman-Rivera, Dhruv Batra, and Pushmeet Kohli. Multiple choice learning: Learning to produce multiple structured outputs. In *Advances in Neural Information Processing Systems*, 2012.

Pouya Hamadanian, Arash Nasr-Esfahany, Malte Schwarzkopf, Siddhartha Sen, and Mohammad Alizadeh. Online reinforcement learning in non-stationary context-driven environments. In *International Conference on Learning Representations*, 2025.

Yizhou Huang, Kevin Xie, Homanga Bharadhwaj, and Florian Shkurti. Continual model-based reinforcement learning with hypernetworks. In *IEEE International Conference on Robotics and Automation*, 2021. arXiv:2009.11997.

Klemens Iten, Bruce Lee, Chenhao Li, Lenart Treven, Andreas Krause, and Bhavya Sukhija. Model-based reinforcement learning for control under time-varying dynamics. *arXiv preprint arXiv:2604.02260*, 2026.

Rituraj Kaushik, Timothée Anne, and Jean-Baptiste Mouret. Fast online adaptation in robotics through meta-learning embeddings of simulated priors. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020.

Samuel Kessler, Mateusz Ostaszewski, Michał Bortkiewicz, Mateusz Żarski, Maciej Wołczyk, Jack Parker-Holder, Stephen J. Roberts, and Piotr Miłoś. The effectiveness of world models for continual reinforcement learning. In *Conference on Lifelong Learning Agents*, 2023. arXiv:2211.15944.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pp. 6405–6416, 2017.

Kimin Lee, Younggyo Seo, Seunghyun Lee, Honglak Lee, and Jinwoo Shin. Context-aware dynamics model for generalization in model-based reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5757–5766, 2020.

Stefan Lee, Senthil Purushwalkam Shiva Prakash, Michael Cogswell, Viresh Ranjan, David Crandall, and Dhruv Batra. Stochastic multiple choice learning for training diverse deep ensembles. In *Advances in Neural Information Processing Systems*, 2016.

Anqi Peter Li and Kaden Kim. Measuring the value of world-model updates: A counterfactual utility protocol for continual adaptation. *arXiv preprint arXiv:2609.10954*, 2026.

Zichen Liu, Guoji Fu, Chao Du, Wee Sun Lee, and Min Lin. Continual reinforcement learning by planning with online world models. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025. arXiv:2507.09177.

Anusha Nagabandi, Ignasi Clavera, Simin Liu, Ronald S. Fearing, Pieter Abbeel, Sergey Levine, and Chelsea Finn. Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. In *International Conference on Learning Representations*, 2019a.

Anusha Nagabandi, Chelsea Finn, and Sergey Levine. Deep online learning via meta-learning: Continual adaptation for model-based RL. In *International Conference on Learning Representations*, 2019b. arXiv:1812.07671.

Kumpati S. Narendra and Jeyendran Balakrishnan. Adaptive control using multiple models. *IEEE Transactions on Automatic Control*, 42(2):171–187, 1997.

Kate Rakelly, Aurick Zhou, Chelsea Finn, Sergey Levine, and Deirdre Quillen. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5331–5340, 2019.

Matteo Saveriano, Yuchao Yin, Pietro Falco, and Dongheui Lee. Data-efficient control policy search using residual dynamics learning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4709–4715, 2017.

Jonathan Schwarz, Jelena Luketina, Wojciech M. Czarnecki, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress & compress: A scalable framework for continual learning. In *Proceedings of the 35th International Conference on Machine Learning*, 2018. arXiv:1805.06370.

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. arXiv:2010.13303.

Bhavya Sukhija, Lenart Treven, Carmelo Sferrazza, Florian Dörfler, Pieter Abbeel, and Andreas Krause. SOMBRL: Scalable and optimistic model-based RL. *arXiv preprint arXiv:2511.20066*, 2025.

Yixiang Sun, Haotian Fu, Michael Littman, and George Konidaris. Knowledge retention for continual model-based reinforcement learning. *arXiv preprint arXiv:2503.04256*, 2025. Also presented at the International Conference on Machine Learning, 2025.

Chi Wan, Kangrui Wang, Yuan Si, Pingyue Zhang, Huang Huang, and Manling Li. WorldAgen: Unified state-action prediction with test-time world model training. In *Proceedings of the Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence*, volume 40, pp. 18584–18592, 2026.

Annie Xie, James Harrison, and Chelsea Finn. Deep reinforcement learning amidst continual structured non-stationarity. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 11393–11403, 2021.

Mengdi Xu, Wenhao Ding, Jiacheng Zhu, Zuxin Liu, Baiming Chen, and Ding Zhao. Task-agnostic online reinforcement learning with an infinite mixture of Gaussian processes. In *Advances in Neural Information Processing Systems*, volume 33, 2020. arXiv:2006.11441.