Temporal Task Diversity: Inductive Biases Under Non-Stationarity in Synthetic Sequence Modelling

Afiq Abdillah Effiezal Aswadi [*1] **Oliver Britton** [*2] **Ross Baker** [*2] **Matthew Farrugia-Roberts** [2]

Abstract

Modern deep learning science often assumes that neural networks learn from a fixed data distribution. However, many practically important learning problems involve data distributions that change throughout training. How does such *non-stationarity* impact the inductive biases of deep learning towards models with different structural, generalisation, and safety properties? A fruitful testbed for studying inductive bias is in-context linear regression sequence modelling, where small transformers display strikingly different generalisation patterns depending on the diversity of the (fixed) training task distribution. In this paper, we explore the effect of diversifying the task distribution *across training time,* finding that such temporal diversity leads to an increased bias towards generalisation over memorisation.

1. Introduction

The science of deep learning aims to understand phenomena exhibited by deep neural networks. As an example phenomenon, transformers trained on linear regression data learn different in-context learning algorithms depending on how many latent regression vectors (or *tasks*) they encounter during training (the *task diversity*; Raventós et al., 2023):

- With low task diversity, the transformer learns a specialised algorithm for distinguishing between the specific latent tasks encountered during training.
- With high task diversity, the transformer learns a general algorithm resembling ridge regression that accommodates both training tasks and novel tasks.

This *task diversity threshold* phenomenon, and other phenomena in similar synthetic sequence modelling settings, have enabled us to study deep learning's so-called *inductive biases*. For example, Hoogland et al. (2025) and Carroll et al. (2025) cast these phenomena as neural networks navigating a trade-off between training loss and model complexity.

Prior work in this vein has considered independently and identically distributed sequence data. However, many important learning problems involve data that is *non-identically* distributed. For example, modern foundation models undergo multiple stages of pre- and post-training, all with different data distributions (Ouyang et al., 2022). Moreover, curriculum learning methods deliberately alter the data distribution throughout training (Bengio et al., 2009), and on-policy (multi-agent) reinforcement learning algorithms generate experience data using the latest policies (Sutton & Whitehead, 1993; Papoudakis et al., 2019). Finally, due to the dynamic nature of our world, many data generating processes encountered in practice evolve over time (Clements & Hendry, 1999; Milly et al., 2008; Nestor et al., 2019).

The introduction of non-stationarity has the potential to alter the way deep learning selects between models with different structures and behaviours. Understanding the principles governing this selection is key to ensuring the safety of future advanced learning systems (Wentworth, 2021; Pepin Lehalleur et al., 2025). With this motivation in mind, we offer the following contributions:

- We extend the synthetic sequence modelling problem of Raventós et al. (2023) to allow for a dynamically varying distribution of latent tasks (Section 3).
- We show experimentally that increasing the rate of change of the task distribution decreases the task diversity threshold (Section 4), below which the transformer tracks the changing set of latent tasks (Section 5).
- We discuss possible explanations of this phenomenon, including that learning is biased towards both model simplicity and, newly, model *stability* (Section 6).

Our results suggest that non-stationary deep learning exhibits an inductive bias of a somewhat richer nature than that exhibited in stationary learning, motivating further study into the extent and mechanism of this phenomenon in this and similar deep learning settings.

[*]Equal contribution (random order). Cite as: Effiezal Aswadi et al., 2026. [1]Independent [2]University of Oxford. Correspondence to: MFR <matthew@far.in.net>.

Published as a paper at the 1st FoGen workshop, ICML 2026, Seoul, South Korea, 2026.

2. Related work

Inductive biases in synthetic sequence modelling. Machine learning problems are often underspecified, meaning that the provided data do not identify a unique solution model (Breiman, 2001, §8). Inductive biases refer to the combined effects of the architecture and learning algorithm in determining the model chosen by a learning process (Mitchell, 1980). In modern deep learning, transformers trained on synthetic sequence modelling data (Garg et al., 2022) have furnished striking examples of critical thresholds at which the solutions found by the learning process change. Examples of such phenomena include the emergence of in-context learning given data sets with certain properties (Chan et al., 2022), or a transition between specialised and generic in-context learning algorithms with increasing task diversity (Raventós et al., 2023; He et al., 2024; Park et al., 2025). To the best of our knowledge, no prior work has studied the inductive biases of *non-stationary* synthetic sequence modelling.

Learning from non-stationary data. Non-stationarity is core to the problems studied in the literatures on lifelong/continual learning (Schlimmer & Fisher, 1986; Thrun & Mitchell, 1995; Chen & Liu, 2018; Parisi et al., 2019; Wang et al., 2024); concept drift or non-stationary learning (Widmer & Kubat, 1996; Tsymbal, 2004; Ditzler et al., 2015); and reinforcement learning (Sutton & Whitehead, 1993; Sutton & Barto, 2018; Papoudakis et al., 2019; Igl et al., 2021). Note that most prior work in these settings aims to engineer architectures or learning algorithms pursuant to various learning objectives. In this work, we instead aim to advance scientific understanding of the phenomena that non-stationarity induces in mainstream architectures and learning algorithms—comparable to work studying phenomena like catastrophic forgetting (McCloskey & Cohen, 1989; Ratcliff, 1990; French, 1999; Goodfellow et al., 2014) or hysteresis (Igl et al., 2021; Nikishin et al., 2022).

3. Problem setting

We study transformers trained on synthetic in-context linear regression data, extending the setting of Raventós et al. (2023) to non-stationary task distributions.

Tasks and sequences. A *task* is a latent regression vector $\mathbf{t} \in \mathbb{R}^D$. Given a task $\mathbf{t}$, define a conditional distribution $q(S \mid \mathbf{t})$ over sequences $S = (x_1, y_1, \ldots, x_K, y_K) \in (\mathbb{R}^D \times \mathbb{R})^K$ by independently sampling inputs $x_k \sim \mathcal{N}(0, I_D)$ and setting $y_k = \mathbf{t}^\top x_k + \varepsilon_k$ with $\varepsilon_k \sim \mathcal{N}(0, \sigma^2)$. Define an unconditional sequence distribution $q(S) = \int q(S \mid \mathbf{t}) q(\mathbf{t}) \mathrm{d}\mathbf{t}$ by first sampling a task from a *task distribution* $q(\mathbf{t})$ and then sampling a sequence from the corresponding conditional distribution.

Following Raventós et al. (2023), we parametrise a family of task distributions $q_M(\mathbf{t})$ by a *task diversity* $M \in \mathbb{N} \cup \{\infty\}$. A task set $\mathcal{T}_M = \{\mathbf{t}_1, \ldots, \mathbf{t}_M\}$ induces the discrete uniform task distribution $q_M(\mathbf{t}) = \mathrm{Uniform}(\mathcal{T}_M)$. We also define $q_\infty(\mathbf{t}) = \mathcal{N}(0, I_D)$.

Non-stationary task distributions. In the stationary setting of Raventós et al. (2023), the task set $\mathcal{T}_M$ is fixed throughout training. We generalise this by allowing the task set to vary with training step $\tau \in \{1, \ldots, T\}$, writing $\mathcal{T}_M(\tau) = \{\mathbf{t}_1(\tau), \ldots, \mathbf{t}_M(\tau)\}$ for the task set at step τ and $q_M^{(\tau)}(\mathbf{t}) = \mathrm{Uniform}(\mathcal{T}_M(\tau))$ for the corresponding task distribution.

Mean squared error objective. Given a sequence S, write $S_{\leq k} = (x_1, y_1, \ldots, x_{k-1}, y_{k-1}, x_k)$ for the *context* when predicting y_k. A predictor f maps contexts to predicted labels. Given a data distribution $q(S)$, define the population loss

$$\ell(f) = \mathop{\mathbb{E}}_{S \sim q} \left[\frac{1}{K} \sum_{k=1}^{K} (f(S_{\leq k}) - y_k)^2 \right]. \tag{1}$$

For a transformer with parameters w, we write $\ell(w)$ for the loss of $f(\cdot; w)$, and $\ell^M(w)$ or $\ell^{M,\tau}(w)$ when the data distribution is q_M or $q_M^{(\tau)}$ respectively.

Optimal predictors. Raventós et al. (2023) showed that the optimal estimator for the k-th prediction, minimising the k-th term in $\ell(f)$, is the Bayesian posterior mean $\hat{y}_k = \hat{\mathbf{t}}_k^\top x_k$ where $\hat{\mathbf{t}}_k = \mathbb{E}[\mathbf{t} \mid S_{\leq k}]$, with the expectation taken over the task distribution $q(\mathbf{t})$ as the prior.

For finite task diversity (prior $q_M(\mathbf{t})$), the posterior given context $X = [x_1 \ \cdots \ x_{k-1}]^\top$ and $\mathbf{y} = (y_1, \ldots, y_{k-1})$ is

$$p(\mathbf{t} \mid X, \mathbf{y}) = \frac{\exp\left(-\frac{1}{2\sigma^2}||\mathbf{y} - X\mathbf{t}||^2\right)}{\sum_{m=1}^{M} \exp\left(-\frac{1}{2\sigma^2}||\mathbf{y} - X\mathbf{t}_m||^2\right)} \tag{2}$$

for $\mathbf{t} \in \mathcal{T}_M$ (and 0 otherwise). The posterior mean then yields the *discrete minimum mean squared error* (dMMSE) predictor, with task estimate

$$\hat{\mathbf{t}}_k^M = \frac{\sum_{m=1}^{M} \exp\left(-\frac{1}{2\sigma^2} \sum_{j=1}^{k-1} (y_j - \mathbf{t}_m^\top x_j)^2\right) \mathbf{t}_m}{\sum_{m=1}^{M} \exp(-\frac{1}{2\sigma^2} \sum_{j=1}^{k-1} (y_j - \mathbf{t}_m^\top x_j)^2)}. \tag{3}$$

For infinite task diversity (prior $q_\infty(\mathbf{t})$), the posterior is a Gaussian with parameters

$$\begin{gathered} \mathbf{t} \mid X, \mathbf{y} \sim \mathcal{N}(\mu_k, \Sigma_k), \\ \Sigma_k = \left(\frac{1}{\sigma^2} X^\top X + I_D\right)^{-1}, \quad \mu_k = \frac{1}{\sigma^2} \Sigma_k X^\top \mathbf{y}. \end{gathered} \tag{4}$$

The posterior mean yields the *ridge* predictor, with task estimate

$$\hat{\mathbf{t}}_k^\infty = (X^\top X + \sigma^2 I_D)^{-1} X^\top \mathbf{y} \tag{5}$$

Thus dMMSE is optimal under q_M for finite M, but is specialised to the particular tasks in $\mathcal{T}_M$, while ridge is optimal only under q_∞ and generalises to any task drawn from $\mathcal{N}(0, I_D)$.

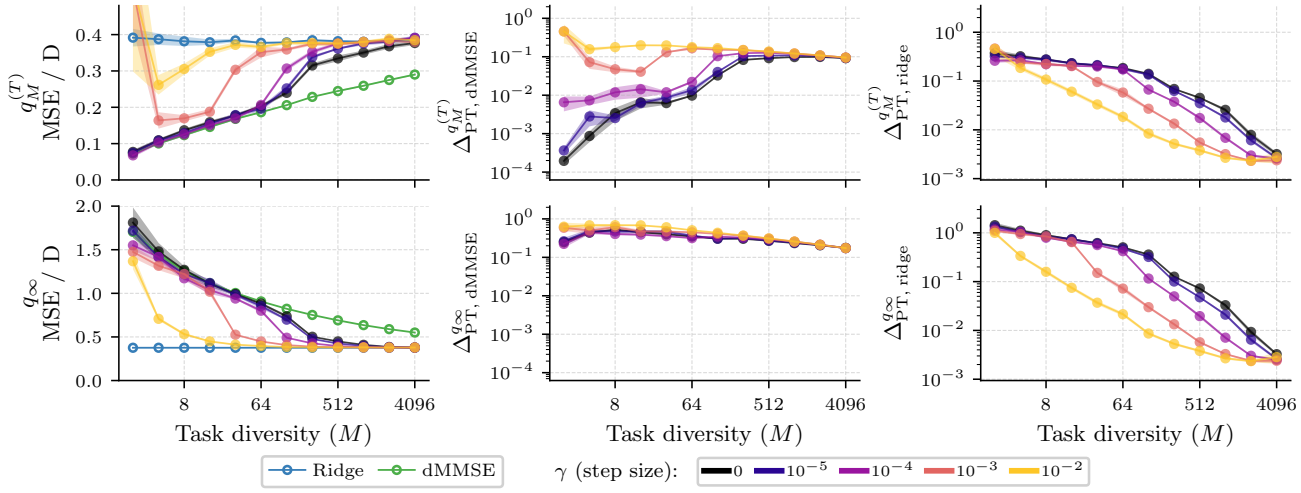

Figure 1. **Increasing non-stationarity via MALA random walk shifts the dMMSE-ridge transition to lower task diversities.** We show results using the final M tasks at the end of pretraining (*top row*) and on new tasks drawn from $\mathcal{T}_{\text{True}} = \mathcal{N}(0, I_D)$ (*bottom row*), for transformers trained via tasks updating according to a MALA random walk with step size γ at each step of training. We vary γ from 0 (dark) to 10^{-2} (light). The *left column* compares the normalised loss of pretrained transformers (PTs) with increasing static task diversity M to that of the dMMSE and ridge reference predictors. The *middle* and *right columns* show the mean squared distance $\Delta_{\text{PT,dMMSE}}$ and $\Delta_{\text{PT,Ridge}}$ between the PT's predictions and those of dMMSE and ridge respectively. Results averaged across 7 seeds, shaded region shows ± 1 standard error. Five outlier runs excluded due to training instability (see Appendix D). At low step sizes, the PT reproduces the results from the stationary setting in Raventós et al. (2023): it approximates dMMSE at low M and transitions to ridge at high M. As γ increases, this transition shifts towards lower task diversities. At extremely low task diversity and large step size, training is unstable.

4. Reducing the task diversity threshold

In this section, we show that increasing the rate of change of the task distribution lowers the task diversity threshold—transformers learning from a more rapidly varying data distribution increasingly prefer the ridge solution. We study two forms of non-stationarity: in Section 4.1, we change the distribution by a small amount every step, and in Section 4.2 we resample tasks at randomly chosen times.

Following Raventós et al. (2023), we train 8-layer transformers using the problem setting described in Section 3 (see Appendix A for details). After training, we compare the predictions of the pretrained transformers (PT) to those of the optimal predictors (dMMSE or ridge) by computing the mean square prediction differences

$$\Delta^{q}_{\text{PT,ref}} = \mathop{\mathbb{E}}_{S\sim q}\left[\frac{1}{KD}\sum_{k=1}^{K}\left(f(S_{\leq k}; w) - \hat{y}_k^{\texttt{ref}}\right)^2\right] \tag{6}$$

where `ref` is either dMMSE or ridge and q is either in-distribution sequences ($q_M^{(\tau)}$) or out-of-distribution sequences (q_∞). These metrics serve as estimates of the corresponding L_2 distances in function space.

4.1. Random walk non-stationarity

First, we introduce non-stationarity by evolving each task vector independently via the Metropolis-Adjusted Langevin Algorithm (MALA; Roberts & Tweedie, 1996), a Markov chain Monte Carlo method that targets $\mathcal{N}(0, I_D)$ as its stationary distribution. We initialise $\mathbf{t}_m(0) \sim \mathcal{N}(0, I_D)$. At each training step τ, each task $\mathbf{t}_m(\tau)$ is updated by first computing a proposal

$$\tilde{\mathbf{t}}_m(\tau+1) = \left(1 - \frac{\gamma}{2}\right)\mathbf{t}_m(\tau) + \sqrt{\gamma}\,\boldsymbol{\xi}_m(\tau) \tag{7}$$

where $\boldsymbol{\xi}_m(\tau) \sim \mathcal{N}(0, I_D)$ and $\gamma > 0$ is the step size. The proposal is then accepted or rejected via a Metropolis-Hastings step, where $\mathbf{t}_m(\tau+1) = \tilde{\mathbf{t}}_m$ with probability α and $\mathbf{t}_m(\tau+1) = \mathbf{t}_m(\tau)$ otherwise, where

$$\alpha = \min\left(1, \exp\left[\frac{\gamma}{8}(||\mathbf{t}_m(\tau)||^2 - ||\tilde{\mathbf{t}}_m||^2)\right]\right). \tag{8}$$

This ensures that $\mathcal{N}(0, I_D)$ is the stationary distribution of the chain so that the marginal distribution of each task at every training step is $\mathcal{N}(0, I_D)$ regardless of γ. The magnitude of the step size controls the rate of change of the task distribution, so that small γ approximates the stationary setting with the task distribution changing only slightly between training steps, while large γ causes the task set to change quickly over the course of training.

We train a separate model for each combination of static task diversity $M \in \{2, 4, 8, \ldots, 4096\}$ and MALA step size $\gamma \in \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$. We also tested larger step sizes ($\gamma \geq 10^{-1}$), but found that training was unstable and the transformer converged to neither dMMSE nor ridge.

The results are shown in Figure 1. At step size 0, we recover the stationary results of Raventós et al. (2023), where

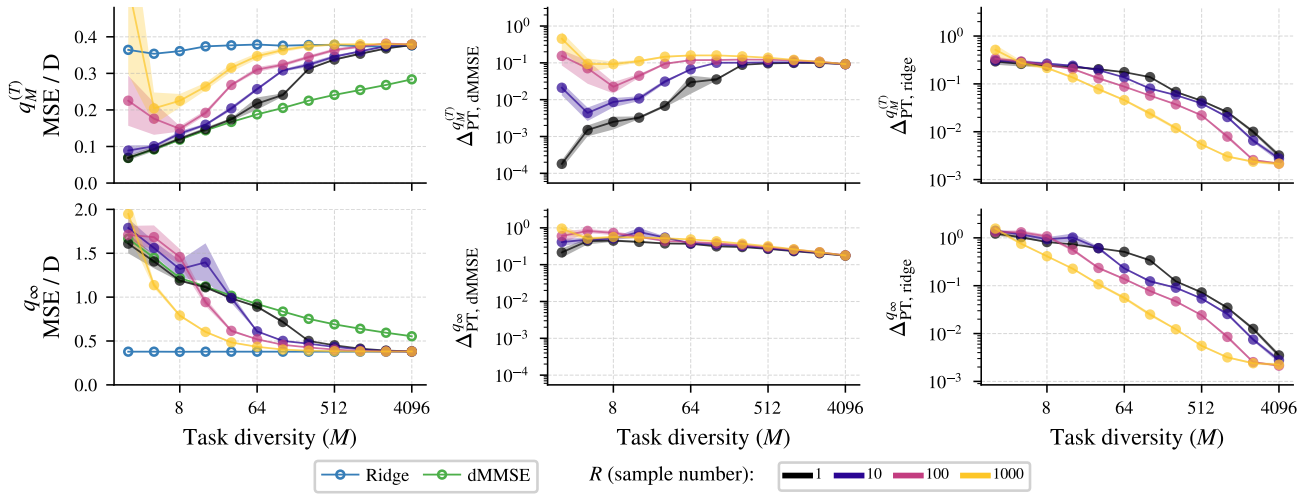

Figure 2. **Increasing non-stationarity via resampling shifts the dMMSE-ridge transition to lower task diversities.** As in Figure 1, we show results for both the final M tasks at the end of pretraining (*top row*) and for new tasks drawn from $\mathcal{T}_{\text{True}} = \mathcal{N}(0, I_D)$ (*bottom row*). We use transformers pretrained with resampling non-stationarity, for which we vary the sample number R from 1 (dark) to 1000 (light). The *left column* compares the normalised loss of the pretrained transformers (PTs) to that of the dMMSE and ridge reference predictors. The *middle* and *right columns* show the mean squared distances $\Delta_{\text{PT,dMMSE}}$ and $\Delta_{\text{PT,Ridge}}$ between the predictions of PT and those of dMMSE and ridge respectively. Note that these dMMSE and ridge predictions do not depend on R. Results averaged across 7 seeds, shaded regions show ± 1 standard error. Two outlier runs excluded due to training instability (see Appendix D). We see again that the transition from dMMSE to ridge shifts to lower task diversities as the sample number R increases.

varying the task diversity causes the transformers to transition from approximating dMMSE at low values of M to approximating ridge at high values of M. As we introduce non-stationarity by increasing γ, this transition occurs at lower task diversities. For the lowest task diversity and largest step sizes, training is unstable (see Appendix D).

4.2. Resampling non-stationarity

We also test the effect of gradually updating the task set $\mathcal{T}_M(\tau) = \{\mathbf{t}_1(\tau), \ldots, \mathbf{t}_M(\tau)\}$ throughout training by resampling each task at $R-1$ randomly selected training steps. In particular, each task is given by a stepwise function

$$\mathbf{t}_i(\tau) = \sum_{j=0}^{R-1} \mathbb{1}_{[\tau_{i,j}, \tau_{i,j+1})}(\tau) \mathbf{t}_{i,j} \quad \forall i \in \{1, ..., M\} \tag{9}$$

where $\mathbf{t}_{i,j} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I_D)$. Each initial sample happens at training step $\tau_{i,0} = 1$, and the updates occur at steps

$$\tau_{i,j} = T \sum_{k=1}^{j} \delta_{i,k} \quad \forall j \in \{1, ..., R-1\} \tag{10}$$

where $(\delta_{i,1}, \ldots, \delta_{i,R}) \overset{\text{i.i.d.}}{\sim} \text{Dir}(\mathbf{1}_R)$. Here we use a Dirichlet distribution since it has the property that $\sum_{j=1}^{R} \delta_{i,j} = 1$, allowing each $\delta_{i,j}$ to represent a proportion of the training run. Finally we let $\tau_{i,R} = T + 1$ so that $\mathbf{t}_i(\tau)$ is defined up to and including the final training step.

The sample number R controls how often the tasks are resampled, with $R = 1$ corresponding to the stationary setting and large R corresponding to frequent task set updates. We designed this setup so that the tasks are updated asynchronously and at irregular intervals, imitating how a data distribution might naturally vary over time. Note that the model sees a total of MR realised task vectors over the course of training.

We train a separate model for each combination of task diversity $M \in \{2, 4, 8, \ldots, 4096\}$ and sample number $R \in \{1, 10, 100, 1000\}$. The results are shown in Figure 2. As usual, we see transformers trained with low M approximating dMMSE, while those trained with high M approximate ridge. For $R = 1$ we recover the stationary results of Raventós et al. (2023), and as R increases the transition between dMMSE and ridge shifts to lower task diversities. Once again, for the lowest task diversities and largest sample numbers, training is unstable.

5. Continual specialisation

Section 4 shows the effect of increasing non-stationarity on whether the *fully-trained* transformer approximates dMMSE or ridge. In this section, we study the evolution of the transformer's behaviour *throughout training* from multiple perspectives. We find that those transformers that eventually approximate dMMSE do so from early in training and continue to track dMMSE as the task distribution changes.

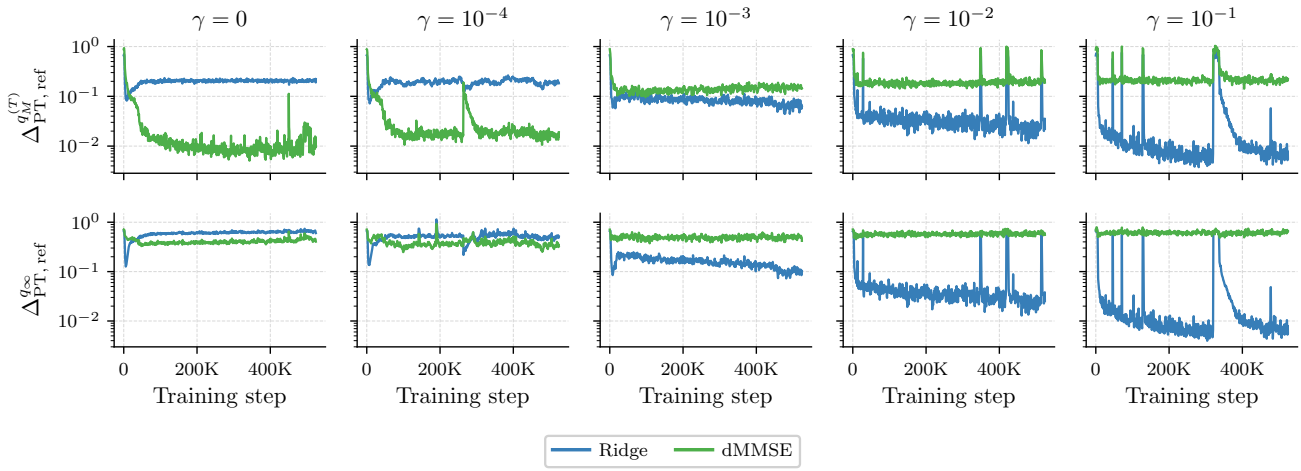

Figure 3. **Predictions of transformers below the non-stationary task diversity threshold track dMMSE throughout training.** We show the mean squared distance $\Delta_{\text{PT,dMMSE}}$ and $\Delta_{\text{PT,Ridge}}$ throughout training. We use a fixed task diversity $M = 32$ and MALA step sizes $\gamma \in \{0, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$. We evaluate on in-distribution sequences from $q_M^{(\tau)}$ (*top row*) and on out-of-distribution sequences from q_∞ (*bottom row*). The results are from a single training seed. The first two columns correspond to transformers below the task diversity threshold for their respective step sizes, and we see that in terms of in-distribution mean squared distance their predictions track the moving dMMSE reference predictor throughout most of training. See Figure 6 for the resampling non-stationarity analogue.

5.1. Perspectives on transformer behaviour

We analyse our transformers' behavioural evolution by measuring distances from the reference predictors in two spaces.

1. **Prediction space** (Section 5.2)**:** We track the mean square difference $\Delta^q_{\mathrm{PT},\mathtt{ref}}$ between the predictions of each transformer checkpoint and each reference predictor (6).
2. **Implicit prior space** (Sections 5.3 and 5.4)**:** We estimate the transformers' implicit prior over latent regression tasks using predictive Monte Carlo (Fortini & Petrone, 2023; 2025; Effiezal Aswadi et al., 2026). We track energy distance (Székely, 1989; Székely & Rizzo, 2013) between these distributions and those of the reference predictors.

Predictive Monte Carlo requires the model to output a full predictive distribution rather than a traditional point prediction for $\hat{y}_k$. Therefore, for the results in this section, we train with a variant architecture replacing the transformer's point-prediction head with an affine transform that outputs the parameters of a mixture of Gaussians. At each query position, the model outputs mixture weights $\{\pi_g\}_{g=1}^G$, means $\{\mu_g\}_{g=1}^G$, and variances $\{\sigma_g^2\}_{g=1}^G$, defining the predictive distribution

$$f(y \mid S_{\le k}) = \sum_{g=1}^{G} \pi_g \, \mathcal{N}(y; \mu_g, \sigma_g^2). \tag{11}$$

We fix the number of mixture components $G = 4$ across all models. We train by minimising the negative log-likelihood (Bishop, 1994), using a logsumexp trick for numerical stability (see Appendix B for details). We also train on longer sequences ($K = 64$ in-context examples) to provide sufficient rollout length for predictive Monte Carlo.

5.2. Prediction trajectories

In this section, we use the random walk non-stationarity setting of Section 4.1 with MALA step size $\gamma \in \{0, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$. We also focus on a fixed task diversity of $M = 32$. This task diversity lies below the task diversity threshold in the stationary setting, but above the threshold for sufficiently high γ. It is therefore interesting to see how different values of γ affect the model's behaviour throughout training. See Appendix C for analysis of resampling non-stationarity and all task diversities.

Figure 3 shows the evolution of a transformer's behaviour throughout training. At low step sizes ($\gamma \in \{0, 10^{-4}\}$) the transformer's predictions track those of the dMMSE reference predictor. At larger step sizes ($\gamma \in \{10^{-3}, 10^{-2}, 10^{-1}\}$), we see the model transitioning to predicting more like the ridge predictor.

For most of training the mean squared differences are approximately stable for each model. Near the beginning of training we see some models briefly trend towards ridge predictions before pivoting back towards dMMSE. This parallels the transient ridge phenomenon (Panwar et al., 2024; Carroll et al., 2025, see also Singh et al., 2024).

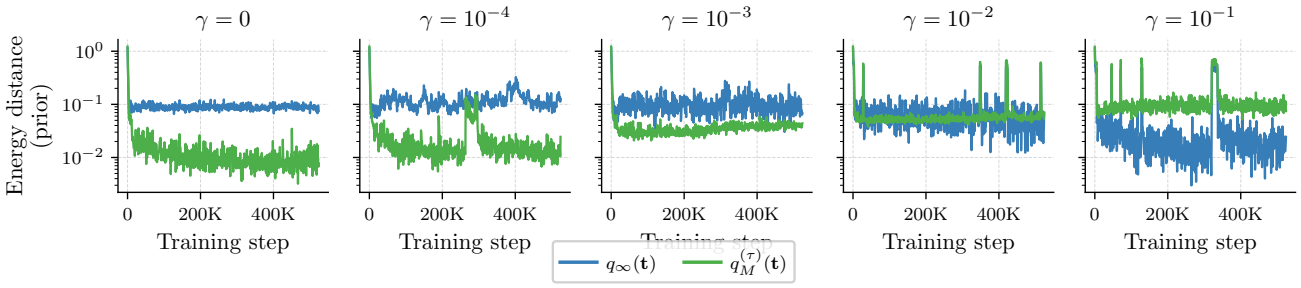

Figure 4. **Predictive Monte Carlo reveals that below the non-stationary task diversity threshold, the transformer's implicit task distribution tracks the changing task distribution.** We use predictive Monte Carlo to extract the transformer's implicit prior over task vectors throughout training, and compare to the finite task distribution $q_M^{(\tau)}(\mathbf{t})$ and infinite task diversity $q_\infty(\mathbf{t}) = \mathcal{N}(0, I_D)$ priors via energy distance. As in Figure 3, we fix $M = 32$ and vary the MALA step size $\gamma \in \{0, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$. The results are from a single training seed. The first two columns correspond to transformers below the task diversity threshold for their respective step sizes, and we see that the revealed priors over tasks closely track the changing task distribution for most of training. See Figure 7 for the resampling non-stationarity analogue.

5.3. Predictive Monte Carlo

Predictive Monte Carlo allows us to sample from the implicit prior and posterior of a Bayes-filtered transformer, as shown in Effiezal Aswadi et al. (2026). We briefly review this methodology, following Fortini & Petrone (2023; 2025). In a supervised setting, a *predictive rule* is a sequence of conditional distributions $f_k(\cdot) = p(y_{k+1} \in \cdot \mid S_{\leq k+1})$. Under (sufficient but not necessary) regularity conditions (Fortini & Petrone, 2023; Fong et al., 2023; Battiston & Cappello, 2025), the predictive distributions converge almost surely along a rollout: $f_L \Rightarrow \tilde{F}$ as $L \to \infty$, where $\tilde{F}$ is a directing measure over observations.

Our transformer provides a predictive rule for labels y_k conditioned on $S_{\leq k}$, but not the input distribution. We therefore construct each rollout by alternating between sampling inputs from the known input distribution and sampling labels from the transformer: for $k = 1, \ldots, L$, we draw $x_k \sim \mathcal{N}(0, I_D)$ and then draw $y_k \sim f(\cdot \mid S_{\leq k})$ from the transformer's predictive distribution.

To recover the transformer's implicit prior over latent tasks, we follow the approach developed by Effiezal Aswadi et al. (2026). For large rollout length L, the generated sequence from a predictive rule $(x_1, y_1), \ldots, (x_L, y_L)$ approximates an i.i.d. sample from $\tilde{F}$. In the linear regression case, $\tilde{F}$ is parametrised by a task vector $\mathbf{t}$. Since OLS is the maximum likelihood estimator, fitting OLS to the generated sequence gives an estimate of the task vector $\mathbf{t}$ that parametrises $\tilde{F}$. Starting from an empty context produces samples from the implicit prior over $\mathbf{t}$. Each rollout converges to a different realisation of $\tilde{F}$, so repeating N times produces a Monte Carlo sample $\{\hat{\mathbf{t}}^{(i)}\}_{i=1}^{N}$ from the transformer's implicit prior over task vectors.

5.4. Implicit prior trajectories

We study the same random-walk non-stationarity setting for task diversity $M = 32$ (see Appendix C for resampling and all task diversities). For predictive Monte Carlo, we sample from the implicit prior of the transformer (rolling out without a prompt). We note that we do not check the conditions sufficient for convergence. We compute energy distance using Monte Carlo samples from the reference priors.

Figure 4 shows that at low step sizes ($\gamma \in \{0, 10^{-4}\}$), below the respective task diversity thresholds, the transformers' revealed priors closely track the task distribution in terms of energy distance. At intermediate step sizes ($\gamma \in \{10^{-3}, 10^{-2}\}$) the revealed prior is somewhere between the instantaneous task distribution $q_{32}^{(\tau)}(\mathbf{t}) = \text{Uniform}(\mathcal{T}_{32}^{(\tau)})$ and the time-average task distribution $q_\infty(\mathbf{t}) = \mathcal{N}(0, I_D)$. At step size $\gamma = 10^{-1}$ the model's implicit prior matches closely the time-average task distribution. These results complement Figure 3 by showing that the same shift occurs in latent task distribution space, not only in prediction space.

Computing energy distance from our reference distributions summarises the implicit prior with a pair of scalars. For a one-dimensional task distribution ($D = 1$), we can visualise the entire implicit prior using a histogram. Figure 5 displays the evolution of this implicit prior histogram over training time for a transformer trained against a single varying task (task diversity $M = 1$, MALA step size $\gamma = 10^{-2}$). We train for only 100K steps, sampling more densely in the 40K–50K interval to observe the distribution's dynamics more clearly. For this configuration, the prior sometimes tracks the changing task, whereas other times it remains concentrated at the origin while the task varies, suggesting we are near a non-stationary task diversity threshold. See Appendix C for all step sizes and resampling non-stationarity.

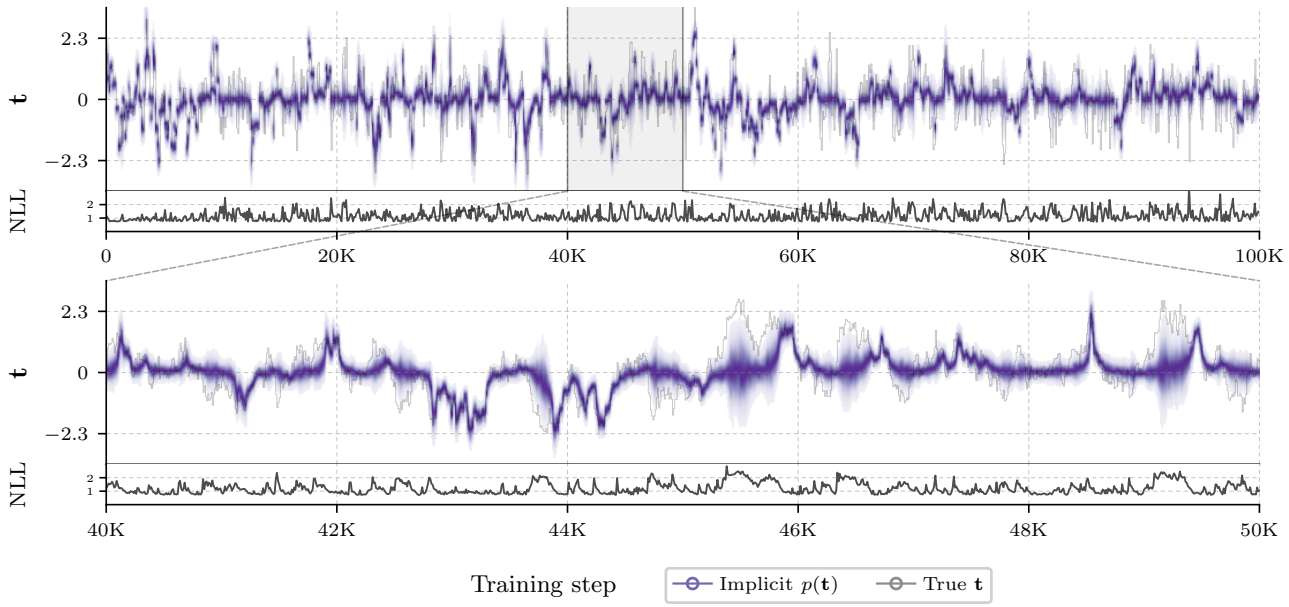

Figure 5. **Implicit prior over a 1D task vector against the true task during training.** We use predictive Monte Carlo to extract the transformer's implicit prior over the task vector $p(\mathbf{t})$ (purple) and compare it to the true task $\mathbf{t}$ (black), for a one dimensional MALA setting with task dimension $D = 1$, task diversity $M = 1$, and $\gamma = 10^{-2}$. The *top panel* shows 0 to 100K training steps, and the *bottom panel* zooms into the 40K to 50K interval. Below each plot we include the (per-token mean) negative log-likelihood on the training batch.

6. Explanations

In this section, we discuss several possible qualitative explanations for the phenomena studied in Sections 4 and 5. We first dismiss two hypotheses that suggest the transformer selects its algorithm based on all tasks seen during training (§6.1, §6.2). We then offer two more nuanced hypotheses, based on diversity amongst tasks seen in recent steps (§6.3) and a bias towards inherently stable solutions (§6.4). Our observations fail to distinguish between these hypotheses, but we propose considerations for future experiments (§6.5).

6.1. Online Bayesian model selection

Bayesian inference is an idealised model of learning and therefore a natural candidate model of deep learning phenomena. In the setting of in-context linear regression specifically, Carroll et al. (2025) and Wurgaft et al. (2025) appeal to Bayesian model selection to explain the dynamic trade-off we observe between dMMSE and ridge solutions.

It is straightforward to model online learning in a Bayesian framework. Suppose we draw data $S_1, \ldots, S_T$ independently from a time-varying distribution $S_\tau \sim q^{(\tau)}$. Given a prior $\pi_0(\theta)$ over some model class, derive a sequence of posteriors using the incremental form of Bayes' rule, $\pi_\tau(\theta) = \mathbb{P}(S_\tau|\theta)\pi_{\tau-1}(\theta)/\mathbb{P}(S_\tau|S_1, \ldots, S_{\tau-1})$. Observe, however: this online formulation is equivalent to full-batch Bayesian inference, $\pi_T(\theta) = \prod_{\tau=1}^{T} \mathbb{P}(S_\tau|\theta)\pi_0(\theta)/\prod_{\tau=1}^{T} \mathbb{P}(S_\tau)$, and therefore invariant to data order. Thus, this model cannot predict a distinction between stationary and non-stationary learning if the long-term time average of the non-stationary distributions matches the stationary distribution.

Both of our non-stationary learning settings have a long-term time average task distribution of $\mathcal{N}(0, I_D)$. Given enough training to realise the long-term time average, this should cause the transformers to always learn the ridge solution (Raventós et al., 2023). Since we observe our transformers tracking the dMMSE solution for hundreds of thousands of steps, we rule out this model.

6.2. Total task diversity

Under the non-stationary condition, our transformers see many more tasks over training than in the stationary setting with the same finite task diversity. Raventós et al. (2023) already showed that transformers that see a larger number of tasks during training are more likely to prefer the ridge solution. Maybe in the non-stationary setting the choice of solution is determined not by the *instantaneous* task diversity M, but by the *total* number of tasks seen throughout training—call the latter the *total task diversity*.

For resampling non-stationarity (§4.2) the number of distinct tasks seen during training is almost surely $R \times M$. For random walk non-stationarity (§4.1), the appropriate effective task diversity is less clear: with small step sizes acceptance probabilities are usually very high but the differences between tasks is extremely small; with sufficiently

large step sizes the distinctions are more significant but acceptance probabilities drop. We could use the integrated autocorrelation time of the random walk as a proxy for the number of steps required to sample a distinct task. Since we are sampling from an 8-dimensional isotropic Gaussian, this quantity should be a small constant.

However, this model predicts much more aggressive lowering of the task diversity threshold than we observe. The total task diversity for many of our training runs exceeds the stationary task diversity threshold, yet we see many examples of these transformers converging to the dMMSE solution. Moreover, these transformers pursue the instantaneous dMMSE throughout training. Therefore, we dismiss the total task diversity hypothesis.

6.3. Effective task diversity within recent memory

In light of the literature on catastrophic forgetting (McCloskey & Cohen, 1989; Ratcliff, 1990; French, 1999; Goodfellow et al., 2014), it is not surprising that our transformers are insensitive to tasks encountered early in training.

We can refine the total task diversity hypothesis by supposing the transformer's selection *is* made based on the stationary task diversity threshold, but against an *effective task diversity* given by the number of tasks seen recently (with the exact meaning of "recently" to be defined). For example, one could posit a fixed number of recent training steps and take the task diversity of the average distribution. Or, one could count all tasks seen but decay their contribution exponentially as they are replaced by new tasks.

For the right definition of effective task diversity, it should be possible to overcome the aggressive predictions of the total task diversity hypothesis.

6.4. Optimisation tends towards more stable models

Finally, we offer an alternative hypothesis—in the presence of non-stationarity, deep learning optimisation trajectories tend towards models that generalise across time and away from models that must be continually adapted in the face of data distribution changes.

The mechanism for this hypothetical tendency would be that internal model structure that contributes towards stable predictions will be continually reinforced during gradient-based training, while structure supporting instantaneous specialisation has to be continually re-learned.

This hypothesis correctly predicts the qualitative shift in task diversity we see: the ridge solution is invariant to the changes in the task distribution we consider, whereas the dMMSE solution repeatedly changes along with the task distribution, forcing a transformer that prefers dMMSE to continually pursue a changing set of tasks.

6.5. Towards distinguishing the latter explanations

The qualitative observation that increasing non-stationarity decreases the task diversity threshold is insufficient to distinguish between the effective task diversity hypothesis and the stability hypothesis. This is a limitation of our experimental design, since increasing non-stationarity coincides with increasing effective task diversity in our set-up.

Future work should design further experiments to isolate the contributions of these (or other) explanations to the observed phenomenon, to the extent possible. We offer some preliminary thoughts in this direction.

1. The challenge is to design an experiment where a change in stability comes apart from a change in effective task diversity. It may be possible to achieve this by considering a data distribution constructed by cycling through a fixed pool of tasks. Cycling tasks makes memorising the current set of tasks unstable, but effective task diversity is capped at the size of the task pool.

2. Alternatively, it may be supposed that, regardless of the precise formalisation of the effective task diversity, the degree of sensitivity of the transformer to recent tasks is dependent on the configuration of the architecture and optimiser (e.g., learning rate) and independent of the choice of data. It may be possible to fit the parameters of a "sensitivity window" at one level of non-stationarity and then see whether these parameters continue to be predictive at another—any discrepancy would point to a contribution from another explanation.

We note that *both* the effective task diversity hypothesis and the stability hypothesis may have roles to play in a full explanation of the phenomenon we have studied. Moreover, they may overlap in that they turn out to offer different perspectives on similar underlying learning dynamics.

7. Conclusion

We have demonstrated that increasing non-stationarity decreases the task diversity threshold for in-context linear regression transformers. Below this reduced task diversity threshold, transformers continuously memorise an ever-changing set of latent regression tasks. Above the reduced task diversity threshold, transformers instead learn the stable ridge solution that generalises across time.

This *temporal task diversity* phenomenon provides a new window into the increasingly important topic of the inductive biases of modern deep learning algorithms, in the important setting of non-stationary deep learning. We invite future work that seeks to more precisely characterise and explain this phenomenon.

Impact statement

This paper presents work whose goal is to advance the science of deep learning by uncovering a novel phenomenon that arises under non-stationarity. While further work is needed to reveal the causes of this phenomenon, we hope that eventually a greater understanding of the nature of the influences of changing data sets on learning in practical settings can contribute to more robust alignment methodologies, along the lines of Pepin Lehalleur et al. (2025).

Reproducibility statement

Experiment details are summarised in Appendix A. For code required to replicate data generation, transformer training, and predictive Monte Carlo methodologies please see github.com/matomatical/temporal-task-diversity.

Acknowledgements

For helpful conversations, we thank Chris Elliott, Lorenz Hufe, Edmund Lau, Daniel Murfet, Susan Wei, and Joan Velja. Research supported with Cloud TPUs from Google's TPU Research Cloud (TRC).

References

Battiston, M. and Cappello, L. Bayesian predictive inference beyond martingales, 2025. Preprint arXiv:2507.21874 [math.ST]. Cited on page 6.

Bengio, Y., Louradour, J., Collobert, R., and Weston, J. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 41–48, 2009. Cited on page 1.

Bishop, C. M. Mixture density networks. Technical Report NCGR/94/004, Neural Computing Research Group, 1994. Cited on pages 5 and 13.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. JAX: composable transformations of Python+NumPy programs. GitHub, 2018. URL http://github.com/jax-ml/jax. Cited on page 12.

Breiman, L. Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical Science*, 16(3):199–231, 2001. Cited on page 2.

Carroll, L., Hoogland, J., Farrugia-Roberts, M., and Murfet, D. Dynamics of transient structure in in-context linear regression transformers, 2025. Preprint arXiv:2501.17745 [cs.LG]. Cited on pages 1, 5, and 7.

Chan, S., Santoro, A., Lampinen, A., Wang, J., Singh, A., Richemond, P., McClelland, J., and Hill, F. Data distributional properties drive emergent in-context learning in transformers. *Advances in Neural Information Processing Systems 35*, pp. 18878–18891, 2022. Cited on page 2.

Chen, Z. and Liu, B. *Lifelong Machine Learning*. Morgan & Claypool, 2018. Cited on page 2.

Clements, M. P. and Hendry, D. F. *Forecasting Non-Stationary Economic Time Series*. MIT Press, 1999. Cited on page 1.

Ditzler, G., Roveri, M., Alippi, C., and Polikar, R. Learning in nonstationary environments: A survey. *IEEE Computational Intelligence Magazine*, 10(4):12–25, 2015. Cited on page 2.

Effiezal Aswadi, A. A., Ma, H., and Wei, S. What does a Bayes-filtered transformer believe? A predictive Monte Carlo approach. In preparation, 2026. Cited on pages 5 and 6.

Elhage, N., Nanda, N., Olsson, C., Henighan, T., Joseph, N., Mann, B., Askell, A., Bai, Y., Chen, A., Conerly, T., DasSarma, N., Drain, D., Ganguli, D., Hatfield-Dodds, Z., Hernandez, D., Jones, A., Kernion, J., Lovitt, L., Ndousse, K., Amodei, D., Brown, T., Clark, J., Kaplan, J., McCandlish, S., and Olah, C. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. Cited on page 12.

Fong, E., Holmes, C., and Walker, S. G. Martingale posterior distributions. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 85(5):1357–1391, 2023. Cited on page 6.

Fortini, S. and Petrone, S. Prediction-based uncertainty quantification for exchangeable sequences. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 381(2247):20220142, 2023. Cited on pages 5 and 6.

Fortini, S. and Petrone, S. Exchangeability, prediction and predictive modeling in Bayesian statistics. *Statistical Science*, 40(1), January 2025. Cited on pages 5 and 6.

French, R. M. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4):128–135, 1999. Cited on pages 2 and 8.

Garg, S., Tsipras, D., Liang, P., and Valiant, G. What can transformers learn in-context? a case study of simple function classes. In *Advances in Neural Information Processing Systems 35*, pp. 30583–30598, 2022. Cited on page 2.

Goodfellow, I. J., Mirza, M., Xiao, D., Courville, A., and Bengio, Y. An empirical investigation of catastrophic forgetting in gradient-based neural networks, 2014. Published as a conference paper at ICLR 2014. Preprint arXiv:1312.6211 [stat.ML]. Cited on pages 2 and 8.

He, T., Doshi, D., Das, A., and Gromov, A. Learning to grok: Emergence of in-context learning and skill composition in modular arithmetic tasks. In *Advances in Neural Information Processing Systems 37*, pp. 13244–13273, 2024. Cited on page 2.

Hoogland, J., Wang, G., Farrugia-Roberts, M., Carroll, L., Wei, S., and Murfet, D. Loss landscape degeneracy and stagewise development in transformers. *Transactions on Machine Learning Research*, 2025. Cited on page 1.

Igl, M., Farquhar, G., Luketina, J., Boehmer, W., and Whiteson, S. Transient non-stationarity and generalisation in deep reinforcement learning. In *International Conference on Learning Representations*, 2021. Cited on page 2.

Karpathy, A. NanoGPT, 2022. URL https://github.com/karpathy/nanoGPT. Cited on page 12.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization, 2015. Published as a conference paper at ICLR 2015. Preprint arXiv:1412.6980 [cs.LG]. Cited on page 12.

McCloskey, M. and Cohen, N. J. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation*, 24: 109–165, 1989. Cited on pages 2 and 8.

Milly, P. C. D., Betancourt, J., Falkenmark, M., Hirsch, R. M., Kundzewicz, Z. W., Lettenmaier, D. P., and Stouffer, R. J. Stationarity is dead: Whither water management? *Science*, 319(5863):573–574, 2008. Cited on page 1.

Mitchell, T. M. The need for biases in learning generalizations. Technical Report CBM-TR-117, Computer Science Department, Rutgers University, 1980. Cited on page 2.

Nestor, B., McDermott, M. B. A., Boag, W., Berner, G., Naumann, T., Hughes, M. C., Goldenberg, A., and Ghassemi, M. Feature robustness in non-stationary health records: Caveats to deployable model performance in common clinical machine learning tasks. In *Proceedings of the 4th Machine Learning for Healthcare Conference*, volume 106, pp. 381–405. PMLR, 2019. Cited on page 1.

Nikishin, E., Schwarzer, M., D'Oro, P., Bacon, P.-L., and Courville, A. The primacy bias in deep reinforcement learning. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pp. 16828–16847. PMLR, 2022. Cited on page 2.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P. F., Leike, J., and Lowe, R. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35*, pp. 27730–27744, 2022. Cited on page 1.

Panwar, M., Ahuja, K., and Goyal, N. In-context learning through the Bayesian prism. In *International Conference on Learning Representations*, 2024. Cited on page 5.

Papoudakis, G., Christianos, F., Rahman, A., and Albrecht, S. V. Dealing with non-stationarity in multi-agent deep reinforcement learning, 2019. Preprint arXiv:1906.04737 [cs.LG]. Cited on pages 1 and 2.

Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., and Wermter, S. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019. Cited on page 2.

Park, C. F., Lubana, E. S., Pres, I., and Tanaka, H. Competition dynamics shape algorithmic phases of in-context learning. In *International Conference on Learning Representations*, 2025. Cited on page 2.

Pepin Lehalleur, S., Hoogland, J., Farrugia-Roberts, M., Wei, S., Gietelink Oldenziel, A., Wang, G., Carroll, L., and Murfet, D. You are what you eat – AI alignment requires understanding how data shapes structure and generalisation, 2025. Preprint arXiv:2502.05475 [cs.LG]. Cited on pages 1 and 9.

Phuong, M. and Hutter, M. Formal algorithms for transformers, 2022. Preprint arXiv:2207.09238 [cs.LG]. Cited on page 12.

Ratcliff, R. Connectionist models of recognition memory: Constraints imposed by learning and forgetting functions. *Psychological Review*, 97(2):285–308, 1990. Cited on pages 2 and 8.

Raventós, A., Paul, M., Chen, F., and Ganguli, S. Pretraining task diversity and the emergence of non-bayesian in-context learning for regression. In *Advances in Neural Information Processing Systems 36*, pp. 14228–14246, 2023. Cited on pages 1, 2, 3, 4, 7, and 12.

Roberts, G. O. and Tweedie, R. L. Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli*, 2(4):341–363, 1996. Cited on page 3.

Schlimmer, J. C. and Fisher, D. A case study of incremental concept induction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 5, 1986. Cited on page 2.

Singh, A., Chan, S., Moskovitz, T., Grant, E., Saxe, A., and Hill, F. The transient nature of emergent in-context learning in transformers. *Advances in Neural Information Processing Systems 36*, pp. 27801–27819, 2024. Cited on page 5.

Sutton, R. S. and Barto, A. G. *Reinforcement Learning: An Introduction*. MIT Press, second edition, 2018. Cited on page 2.

Sutton, R. S. and Whitehead, S. D. Online learning with random representations. In *Proceedings of the Tenth International Conference on Machine Learning*, pp. 314–321. Morgan Kaufmann, 1993. Cited on pages 1 and 2.

Székely, G. J. Potential and kinetic energy in statistics. Lecture notes, Budapest Institute of Technology (Technical University), 1989. As cited in Székely & Rizzo (2013). Cited on page 5.

Székely, G. J. and Rizzo, M. L. Energy statistics: A class of statistics based on distances. *Journal of Statistical Planning and Inference*, 143(8):1249–1272, 2013. Cited on pages 5 and 11.

Thrun, S. and Mitchell, T. M. Lifelong robot learning. *Robotics and Autonomous Systems*, 15(1):25–46, 1995. Cited on page 2.

Tsymbal, A. The problem of concept drift: definitions and related work. Technical Report TCD-CS-2004-15, Department of Computer Science, Trinity College Dublin, 2004. Cited on page 2.

Wang, L., Zhang, X., Su, H., and Zhu, J. A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(8):5362–5383, 2024. Cited on page 2.

Wentworth, J. S. Selection theorems: A program for understanding agents. AI Alignment Forum, 2021. URL https://www.alignmentforum.org/posts/G2Lne2Fi7Qra5Lbuf. Cited on page 1.

Widmer, G. and Kubat, M. Learning in the presence of concept drift and hidden contexts. *Machine Learning*, 23: 69–101, 1996. Cited on page 2.

Wurgaft, D., Lubana, E. S., Park, C. F., Tanaka, H., Reddy, G., and Goodman, N. In-context learning strategies emerge rationally. In *Advances in Neural Information Processing Systems 38*, 2025. Cited on page 7.

A. Experiment details

Architecture. We use a decoder-only transformer with pre-layer normalisation, learnable positional embeddings, and causal masking (see, e.g., Elhage et al., 2021; Phuong & Hutter, 2022). Specifically, we use 8 layers, two attention heads per layer, and an embedding and MLP dimension of 128. A final layer normalisation precedes either a learned affine output head that maps to a scalar prediction in Sections 4.1 and 4.2, or to a mixture density head which maps to the means, variances, and mixture weights of a G-component Gaussian mixture in Section 5.

Input tokenisation. Following Raventós et al. (2023), we encode each sequence $S = (x_1, y_1, \ldots, x_K, y_K)$ as a sequence of $2K$ input tokens in $\mathbb{R}^{D+1}$:

$$\left(\begin{pmatrix}0\\x_1\end{pmatrix}, \begin{pmatrix}y_1\\0\end{pmatrix}, \begin{pmatrix}0\\x_2\end{pmatrix}, \begin{pmatrix}y_2\\0\end{pmatrix}, \ldots, \begin{pmatrix}0\\x_K\end{pmatrix}, \begin{pmatrix}y_K\\0\end{pmatrix}\right).$$

Output tokenisation. We output token sequences of length $2K$, but discard every second token so as to only consider one prediction per regression example. Each output token is either a scalar prediction $\tilde{y}_k$ (for the usual point prediction setting) or $3G$ scalars (for the mixture density setting, see Section 5.1).

Training. We train each transformer for $T = 524$K steps using a batch size of $B = 256$. We optimize with Adam (Kingma & Ba, 2015) without weight decay. We use a learning rate schedule that increases linearly from 0 to its maximum value η over a fraction of training steps and then remains constant. For Sections 4.1 and 4.2, the loss is mean squared error over all K in-context examples. For Section 5, the loss is negative log-likelihood under the mixture density head.

Compute. We use a bespoke transformer implementation modelled after NanoGPT (Karpathy, 2022) and using JAX (Bradbury et al., 2018). Training one model (524K batches of 256 sequences) took approximately 45 minutes on a single dual-core TPU v4 device. Predictive Monte Carlo for one checkpoint (5K rollouts of length 64) took around 25 TPU-device-seconds. Each column in Figure 4 involves 656 checkpoints, thus taking approximately 5 hours (plus transformer training time). TPUs were provided by Google's TPU Research Cloud.

Table 1. **Summary of hyper-parameters.** Where a single hyper-parameter is shared across all experiments, we omit repetitions (—).

Category	Hyper-parameter	§4.1	§4.2	§5
Data	Task dimension (D)	8	—	—
Data	In-context examples (K)	16	16	64
Data	Observation noise variance (σ^2)	0.25	—	—
Model	Number of layers	8	—	—
Model	Number of attention heads per layer	2	—	—
Model	Embedding dimension	128	—	—
Model	MLP hidden dimension	128	—	—
Model	Layer normalisation	Pre	—	—
Model	Prediction head type	Point	Point	Mixture density
Model	Mixture components (G)			4
Optimiser	Type of optimiser	Adam	—	—
Optimiser	Moment estimate decay rates (β_1,β_2)	$(0.9, 0.999)$	—	—
Optimiser	Weight decay	No	—	—
Learning rate schedule	Shape	Increase then constant	—	—
Learning rate schedule	Peak learning rate (η)	3×10^{-3}	—	—
Learning rate schedule	Warm-up strategy	Linear	—	—
Learning rate schedule	Warm-up fraction	10%	—	—
Experiments	Batch size (B)	256	—	—
Experiments	Training steps (T)	524,000	—	—
Experiments	Number of training seeds	7	7	1
Predictive Monte Carlo	Number of rollouts			5,000
Predictive Monte Carlo	Rollout length			64

B. Derivation of mixture density head

We replace the point prediction head with a learned affine map to $\mathbb{R}^{3G}$, parametrising a Gaussian mixture model, as in Bishop (1994). Following the tokenisation scheme described in Appendix A, we keep only the predictions at query positions (every second token). Each prediction vector is partitioned into G separate logits z_g^π, means μ_g, and raw parameters z_g^σ. We map z_g^σ to a positive real number representing the variance using the softplus function, $\sigma_g^2 = \mathrm{softplus}(z_g^\sigma)$. The logits could be converted to probabilities π_g using $\mathrm{softmax}$ though in practice we never use this as we work in log-space in our implementations.

The loss function is the negative log-likelihood of the observed y under the predicted mixture:

$$-\log p(y \mid \pi, \mu, \sigma^2) = -\log \sum_{g=1}^{G} \pi_g \, \mathcal{N}(y;\, \mu_g,\, \sigma_g^2).$$

For numerical stability, we rewrite each term inside the sum as

$$\pi_g \, \mathcal{N}(y;\, \mu_g,\, \sigma_g^2) = \exp(\log \pi_g - \mathrm{NLL}_g)\,,$$

where

$$\mathrm{NLL}_g = \frac{1}{2}\left(\log(2\pi\sigma_g^2) + \frac{(y-\mu_g)^2}{\sigma_g^2}\right)$$

is the (Gaussian) negative log-likelihood for component g. This then gives

$$-\log p(y \mid \pi, \mu, \sigma^2) = -\,\mathrm{logsumexp}_g(\log \pi_g - \mathrm{NLL}_g)\,,$$

which avoids evaluating small exponentials directly.

C. Additional plots

Figures 6 and 7 show the resampling non-stationarity analogues of Figures 3 and 4 from Section 5, exhibiting the same qualitative behaviour.

Figures 8 to 13 show mean squared prediction differences $\Delta_{\text{PT,dMMSE}}$, $\Delta_{\text{PT,Ridge}}$ and the energy distance to the reference priors throughout training for random walk and resampling settings. For each metric we show the full grid of training trajectories with each combination of task diversity M and non-stationarity parameter (γ for random walk, R for resampling). Curves are Gaussian-smoothed with $\sigma = 3$ samples (one sample per 800 training steps). The figures show a decrease in the task diversity threshold as the degree of non-stationarity increases. For low task diversity and low degree of non-stationarity, the model can track the moving task distribution. However, as the task diversity increases, and as the degree of non-stationarity increases, the model struggles to track the moving task distribution, instead differing to the generalising approach.

Figures 14 and 15 show the implicit prior histograms over training time for the dimension $D = 1$, task diversity $M = 1$ setting studied in Section 5.4, but for additional MALA step sizes. We also consider resampling non-stationarity (with uniformly distributed sample events, different from the Dirichlet scheme discussed in Section 4.2). We see that the implicit prior closely tracks the changing tasks in most cases, though less precisely once the non-stationarity increases.

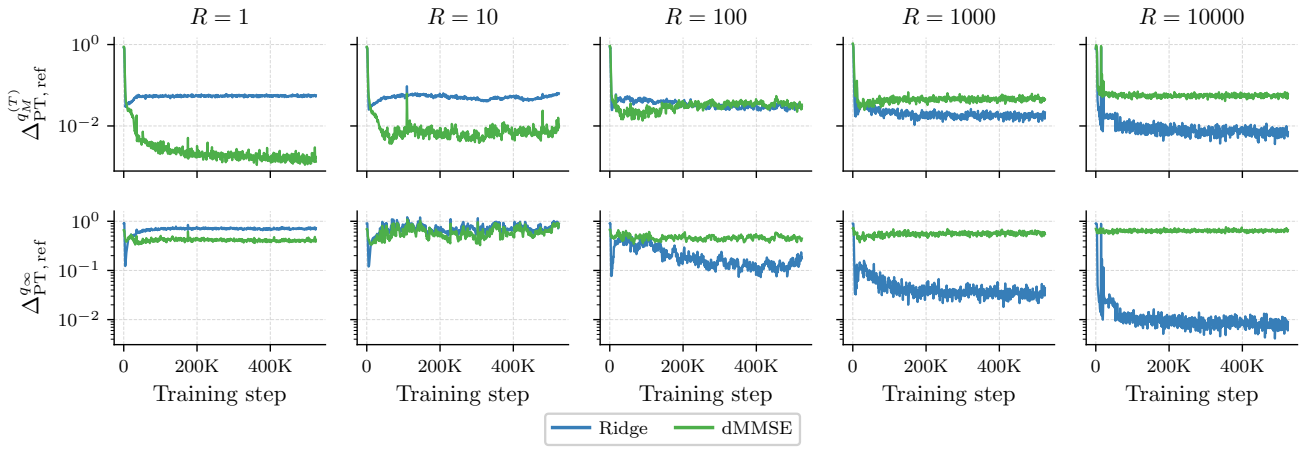

Figure 6. Resampling non-stationarity analogue of Figure 3, for task diversity $M = 32$ and sample numbers $R \in \{1, 10, 100, 1000, 10000\}$. The first two columns correspond to transformers below the task diversity threshold for their respective resampling number R, and we see that in terms of in-distribution mean squared distance their predictions track the moving dMMSE reference predictor throughout most of training.

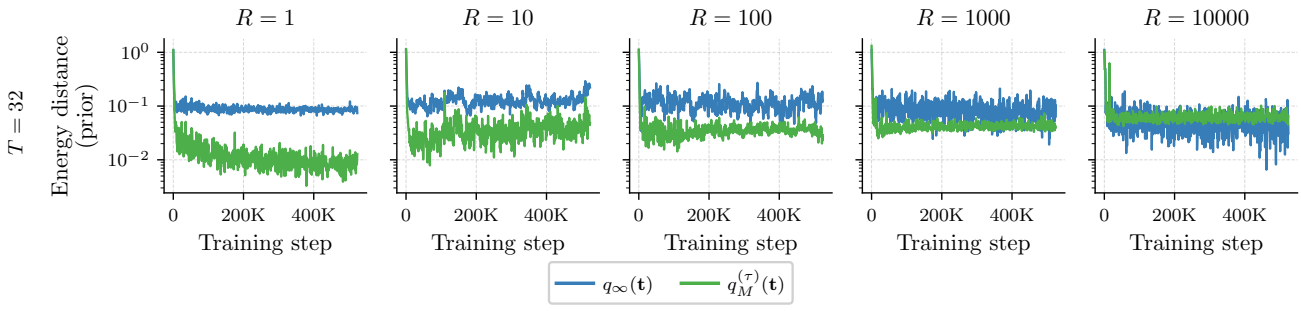

Figure 7. Resampling non-stationarity analogue of Figure 4, for task diversity $M = 32$ and sample numbers $R \in \{1, 10, 100, 1000, 10000\}$. The first two columns correspond to transformers before the task diversity threshold for their respective resampling number R, and we see that the revealed priors over tasks closely track the changing task distribution for most of training.

Figure 8. **In-distribution mean squared prediction differences throughout training under random walk non-stationarity.** We show $\Delta_{\text{PT,dMMSE}}$ and $\Delta_{\text{PT,Ridge}}$ on in-distribution sequences from $q_M^{(\tau)}$ for each combination of task diversity M and MALA step size γ.

Figure 9. **Out-of-distribution mean squared prediction differences throughout training under random walk non-stationarity.** We show $\Delta_{\mathrm{PT,dMMSE}}$ and $\Delta_{\mathrm{PT,Ridge}}$ on out-of-distribution sequences from q_∞ for each combination of task diversity M and MALA step size γ.

Figure 10. **Energy distance between the transformer's implicit prior and the Uniform$(\mathcal{T}_M^{(\tau)})$ and $\mathcal{N}(0, I_D)$ priors throughout training under random walk non-stationarity.** For each combination of task diversity M and MALA step size γ, we use predictive Monte Carlo to extract the transformer's implicit prior over task vectors throughout training, and compare to the baseline priors via energy distance.

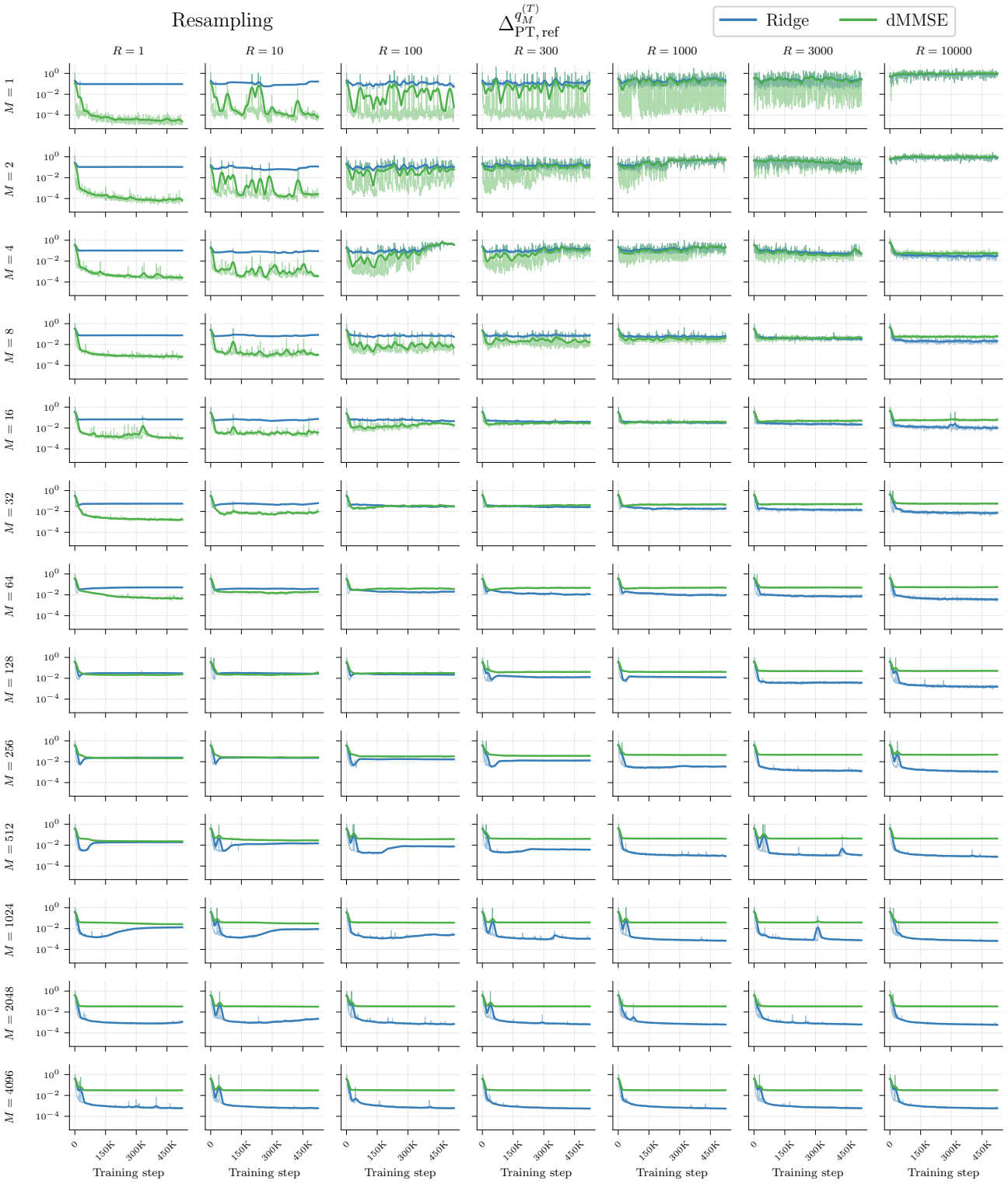

Figure 11. **In-distribution mean squared prediction differences throughout training under resampling non-stationarity.** We show $\Delta_{\text{PT,dMMSE}}$ and $\Delta_{\text{PT,Ridge}}$ on in-distribution sequences from $q_M^{(\tau)}$ for each combination of task diversity M and sample number R.

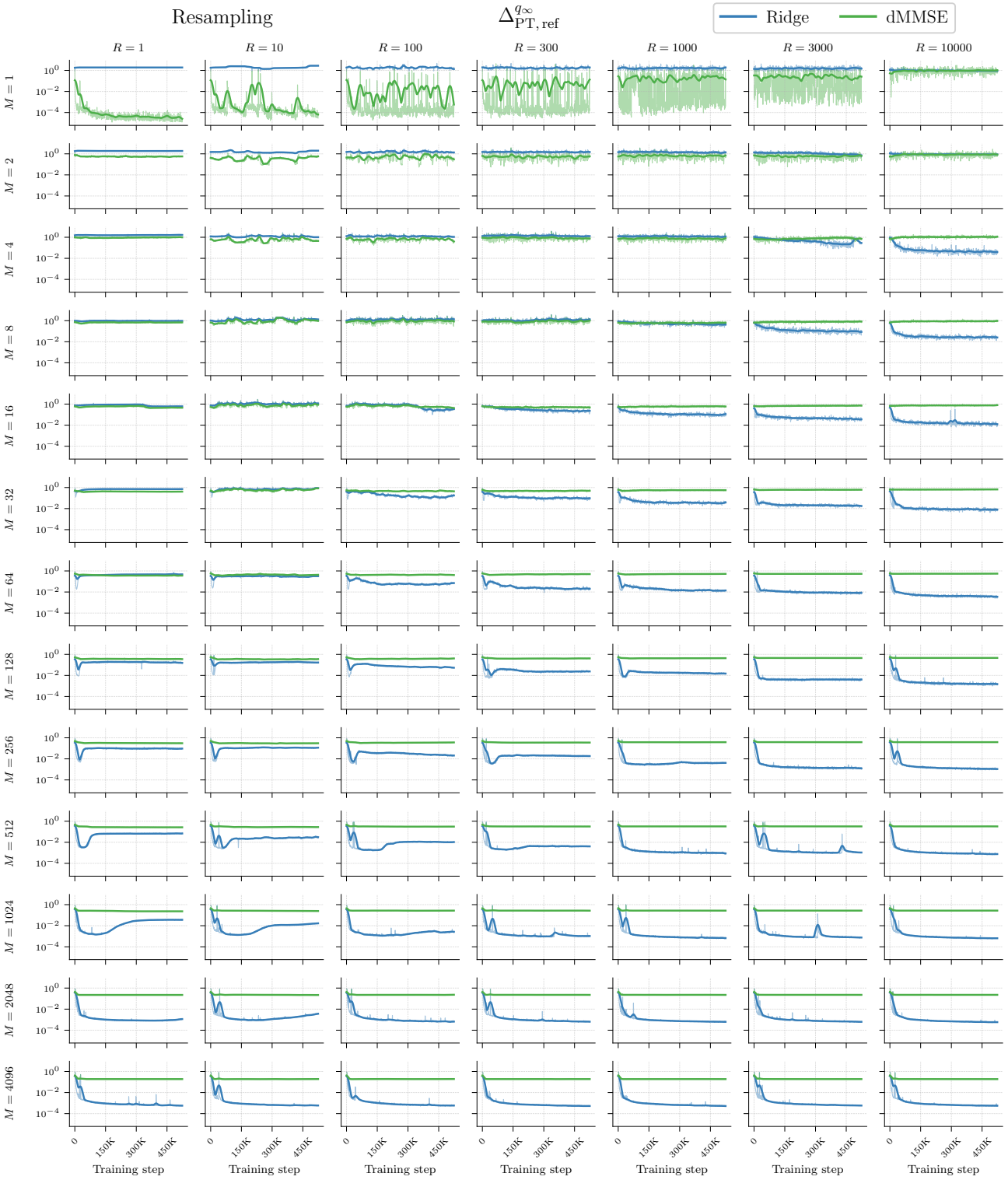

Figure 12. **Out-of-distribution mean squared prediction differences throughout training under resampling non-stationarity.** We show $\Delta_{\text{PT,dMMSE}}$ and $\Delta_{\text{PT,Ridge}}$ on out-of-distribution sequences from q_∞ for each combination of task diversity M and sample number R.

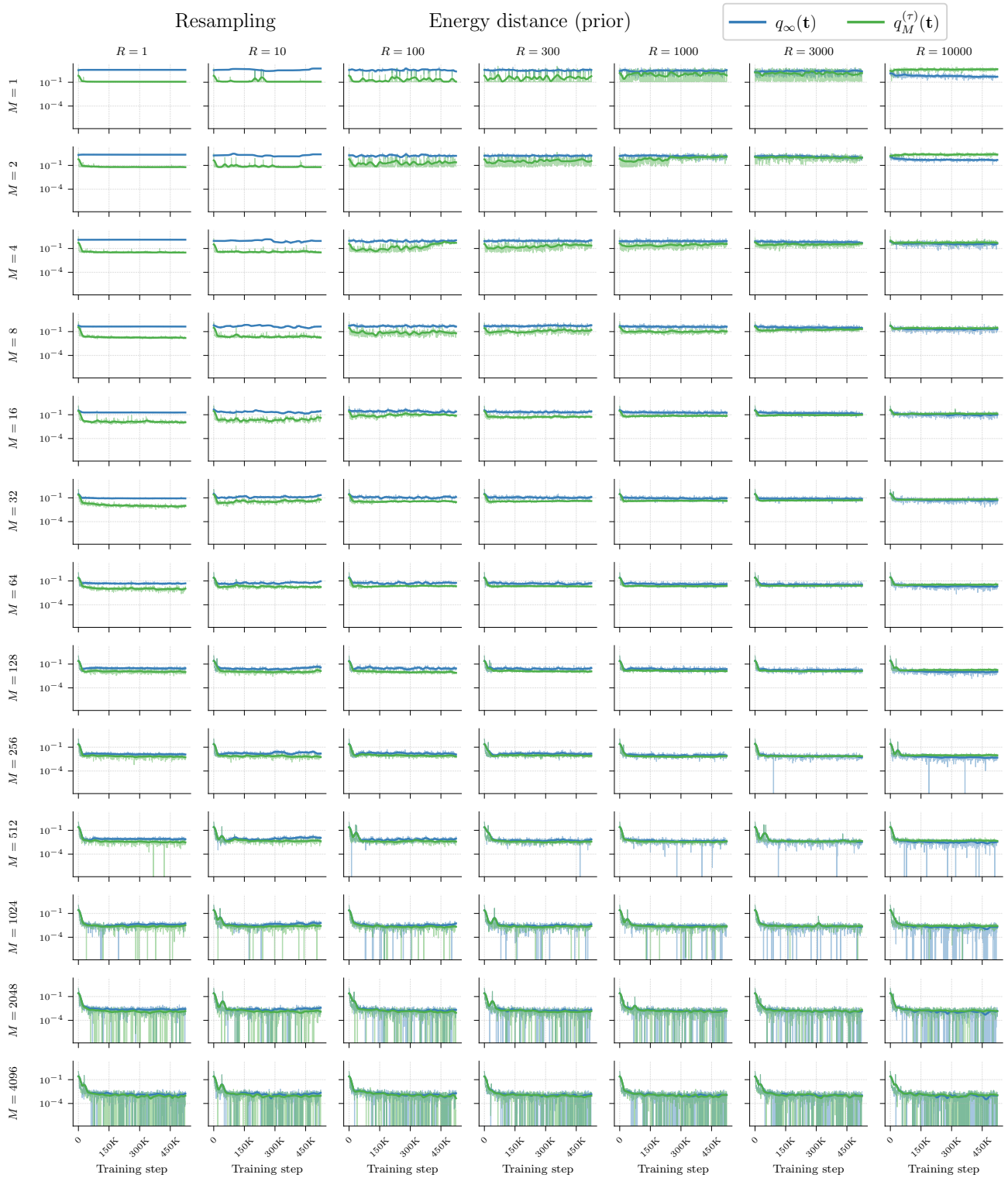

Figure 13. **Energy distance between the transformer's implicit prior and the Uniform$(\mathcal{T}_M^{(\tau)})$ and $\mathcal{N}(0, I_D)$ priors throughout training under resampling non-stationarity.** For each combination of task diversity M and sample number R, we use predictive Monte Carlo to extract the transformer's implicit prior over task vectors throughout training, and compare to the baseline priors via energy distance.

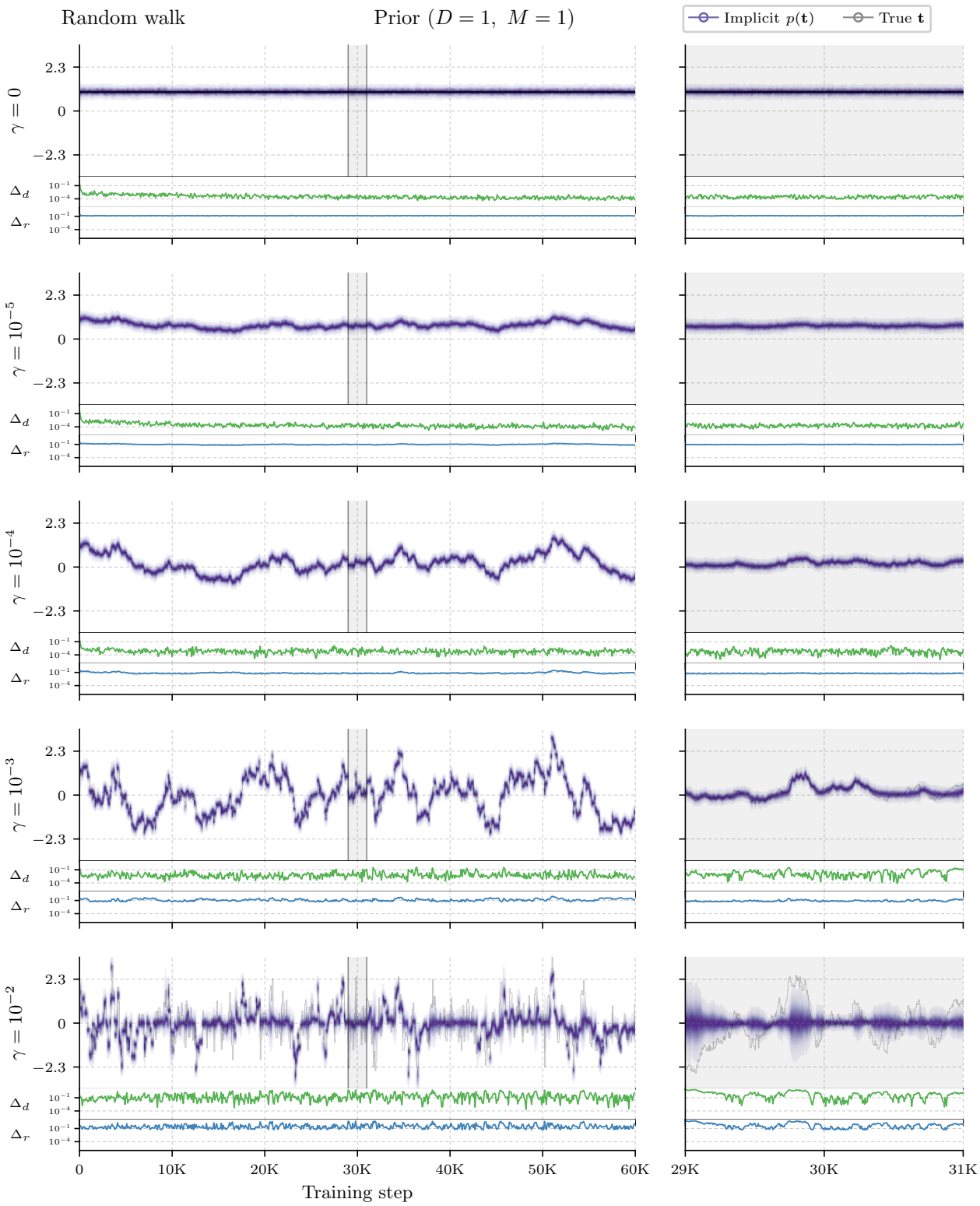

Figure 14. **Implicit prior over a 1D task vector against the true task during training, across MALA step sizes.** As in Figure 5, we use predictive Monte Carlo to extract the transformer's implicit prior over the task vector $p(\mathbf{t})$ (purple) and compare it to the true task $\mathbf{t}$ (black), for a one-dimensional MALA setting with task dimension $D = 1$, task diversity $M = 1$ and $\gamma \in \{0, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$. On the left, we show the task vector over the initial 60K steps of training. On the right, we zoom in on the task vectors during steps 29K–31K. As γ increases, the random walk moves faster and the implicit prior tracks it less precisely. We additionally plot Δ_d and Δ_r over each training step, the mean squared distances between the transformer and the dMMSE and ridge predictors on the training distribution.

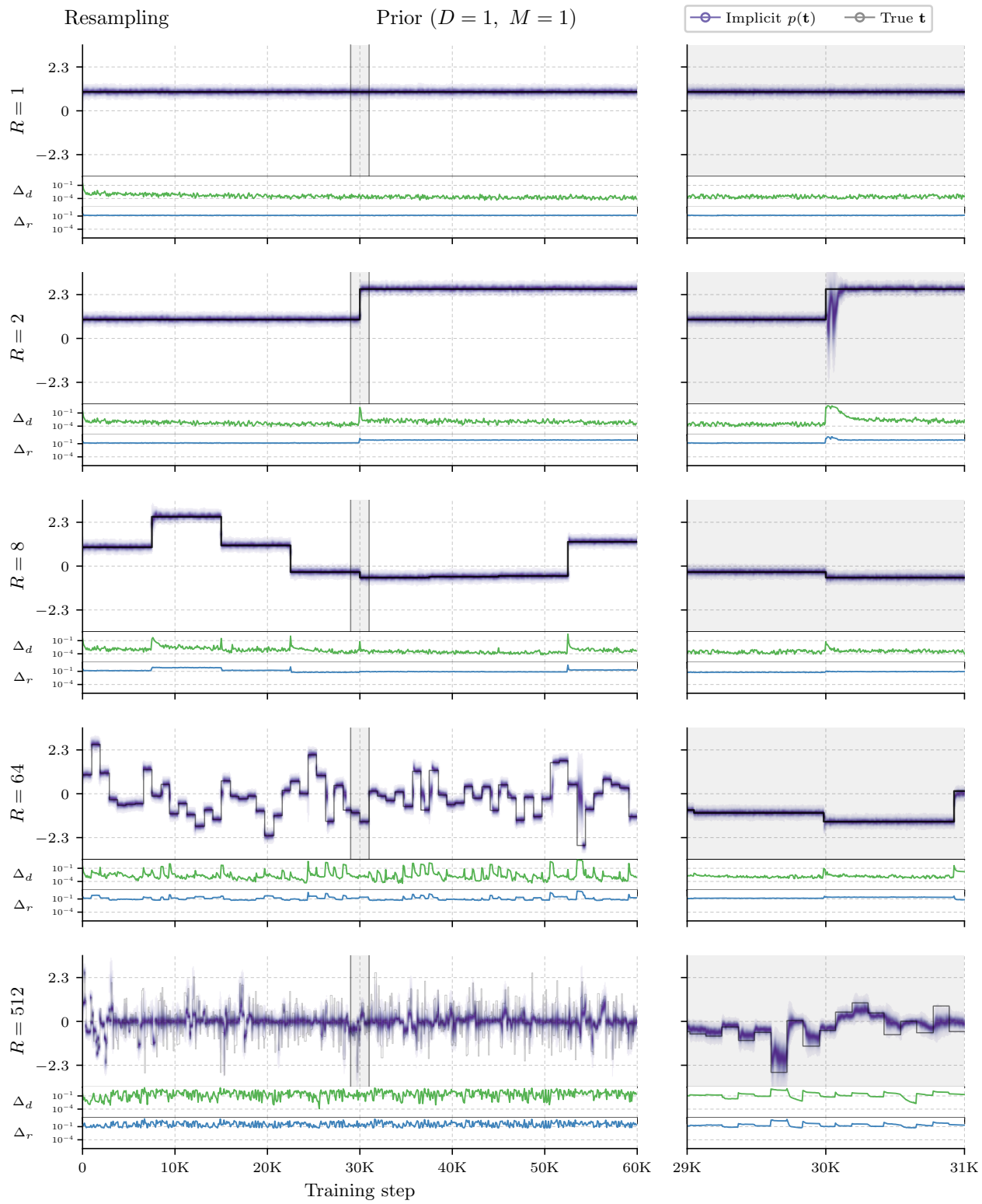

Figure 15. **Implicit prior over a 1D task vector against the true task during training, across resampling rates.** As in Figure 14, we use predictive Monte Carlo to extract the transformer's implicit prior over the task vector $p(\mathbf{t})$ (purple) and compare it to the true task $\mathbf{t}$ (black), for a one-dimensional modified Dirichlet setting where we use R independently-sampled task sets at equally spaced intervals with task dimension $D = 1$, task diversity $M = 1$ and $R \in \{1, 2, 8, 64, 512\}$. On the left, we show the task vector over the initial 60K steps of training. On the right, we zoom in on the task vector during steps 29K-31K. As R increases, the task vector changes more frequently and the implicit prior tracks it less precisely. We also plot Δ_d and Δ_r, the mean squared distances between the transformer and the dMMSE and ridge predictors on the training distribution.

D. Training instability

In Section 4, our initial sweep produced a small number of runs with training stability issues, which we summarise in Figure 16. These broadly fell into two types of issues.

1. The first type of issue was a divergence of loss compared to the other seeds. In five of our runs across the Section 4.1 sweep and two runs across the Section 4.2 sweep, a single seed departed from the trajectory followed by the other seeds in the same configuration. The first two panels of Figure 16 show representative examples. The $(M = 16, \gamma = 0)$ run showed an abrupt spike in loss near the end of training, not seen in the other seeds, while the $(M = 8, \gamma = 10^{-3})$ run drifted slowly upwards starting from around step 400K. We re-ran these outlier seeds with new seeds and used the replacement data in Figures 1 and 2.

2. The second type of instability was a configuration-wide failure to converge. In the low-task-diversity, large-step-size portion of the sweep, every seed oscillated chaotically with training MSE between roughly 1 and 10. The loss in these runs sits near that of ridge, but $\Delta^{q_M^{(\tau)}}_{\text{PT, ridge}}$ remains large, so we cannot interpret these runs as having found the ridge solution. Because no seed in this configuration converged, we did not replace these runs in Figures 1 and 2.

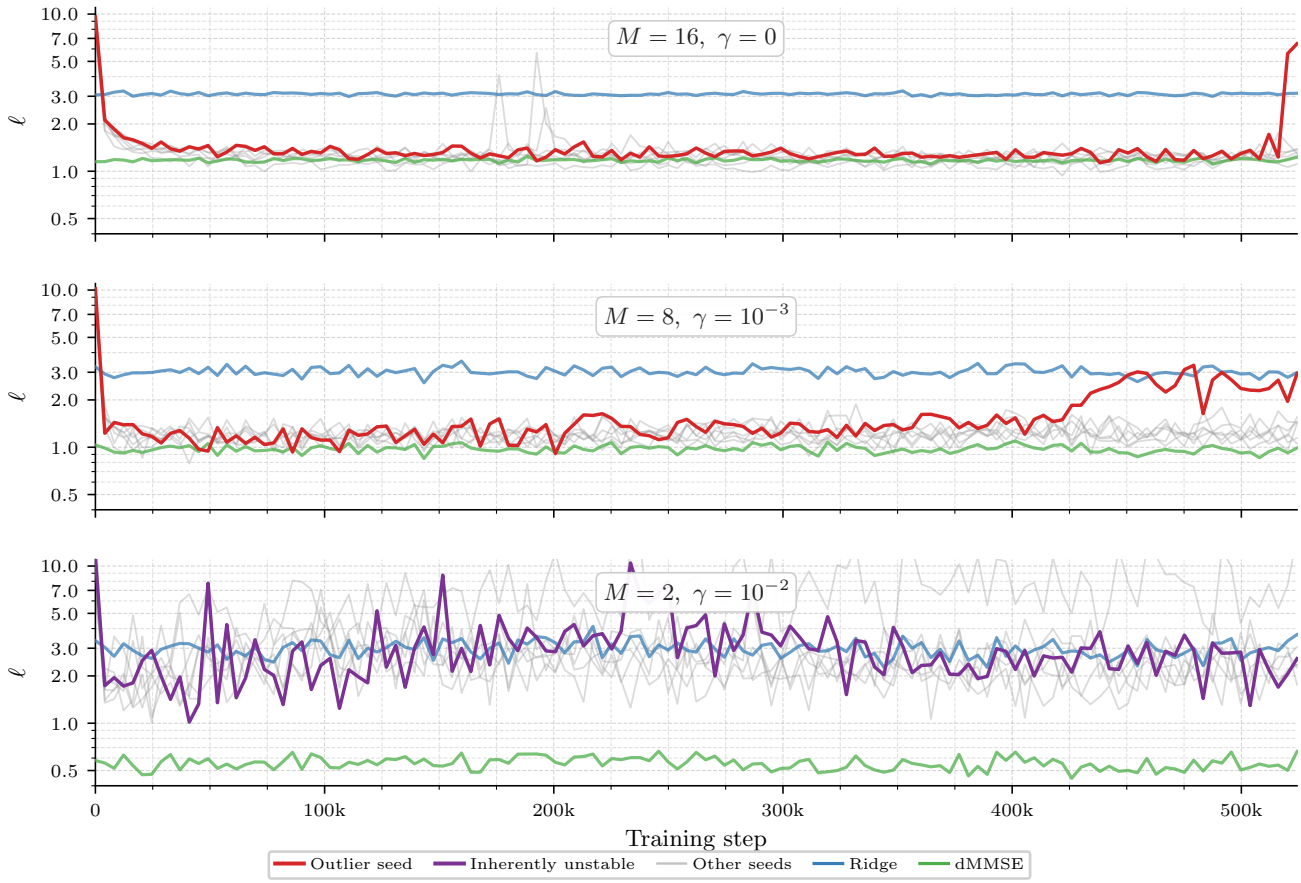

Figure 16. **Examples of training instability.** Each panel shows ℓ_τ^M (log scale) against training step for a single run, evaluated every 4096 steps. In the first two panels we highlight outlier seeds (red) against the other seeds in their configuration (grey). In the third we show an entire configuration that failed to converge (purple). The losses of the ridge and dMMSE predictors are shown in blue and green respectively. The configurations are $(M = 16, \gamma = 0)$, which exhibits a late spike in loss, $(M = 8, \gamma = 10^{-3})$, which drifts upward roughly from step 400K, and $(M = 2, \gamma = 10^{-2})$, for which no seed converged.