LAYER-SELECTIVE UTILITY GATING FOR CONTINUAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural networks in continual learning suffer from plasticity loss. Utility-based perturbed gradient descent (UPGD-W) mitigates this problem through utility-dependent modulation of parameter updates. We investigate how this modulation should be allocated across network layers. We first analyze an idealized continuous-time model and show that, under regularity and stationarity assumptions, its expected displacement-based utility rate relates to gated local curvature with an explicit correction for non-equilibrium probability currents. We then derive a distributional tracking bound that separates the contributions of representation and head drift. These results provide a theoretical motivation for comparing output-only, hidden-only, and full gating across different adaptation settings. On three supervised label-permutation benchmarks, output-only gating applies utility modulation to only 0.16%–2.5% of parameters while achieving 114%–285% higher mean accuracy than Shrink and Perturb and up to 13% higher than full gating. All layers continue to update. The gating variants perform similarly on Input-Permuted MNIST. In Gridworld with controlled dynamics shifts, hidden-only gating achieves higher mean return than output-only gating. Together, these findings show that the benefits of utility gating depend on its placement and the adaptation demands of the learning setting.

1 INTRODUCTION

The goal of continual learning is to develop algorithms capable of learning from non-stationary data distributions without losing the ability to acquire new knowledge (Ring, 1994; Thrun, 1998). While modern neural networks can learn from fixed datasets and generalize to unseen data from the same distribution, they struggle to adapt when the data distribution changes over time. This presents a critical limitation for real-world deployment where environments evolve continuously (Lyle et al., 2023; Dohare et al., 2023; Abbas et al., 2023; Nikishin et al., 2022).

This challenge manifests as two interrelated phenomena, namely loss of plasticity, defined as the progressive inability to learn from new data (Berariu et al., 2021; Lyle et al., 2023; Abbas et al., 2023), and catastrophic forgetting, defined as the abrupt loss of prior knowledge (Kirkpatrick et al., 2017). While forgetting has received substantial attention in the literature, plasticity loss in deep neural networks has attracted increasing attention in recent years (Lyle et al., 2023; Abbas et al., 2023; Nikishin et al., 2022). Proposed contributors to plasticity loss include dead or dormant neurons (Dohare et al., 2023; Sokar et al., 2023), large weight magnitudes (Kumar et al., 2023), feature-rank collapse (Kumar et al., 2021), and the loss of curvature directions (Lewandowski et al., 2023). Plasticity loss also arises in deep reinforcement learning, where changing policies and data distributions create continuing demands for adaptation (Klein et al., 2024; Juliani & Ash, 2024). These demands can differ across network components, motivating interventions that account for where adaptation occurs.

Existing approaches preserve plasticity through perturbation, selective reinitialization, and regularization (Ash & Adams, 2020; Dohare et al., 2024; Kumar et al., 2023). Recent work shows that reinitializing low-utility weights can maintain plasticity in settings where unit reinitialization is less effective (Hernandez-Garcia et al., 2026), while spectral regularization seeks to preserve trainability by controlling layerwise singular values (Lewandowski et al., 2025). UPGD uses utility estimates to modulate gradient updates and perturbations for individual parameters (Elsayed & Mahmood,

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.

2024). These approaches motivate examining both how an intervention operates and where it is applied. For utility gating, we ask whether applying modulation throughout the network is beneficial, or whether restricting it to selected layers better suits the learning setting.

We investigate where utility gating is beneficial using layer-selective variants of UPGD-W (Elsayed & Mahmood, 2024). Our analysis relates displacement-based utility to local curvature and establishes a tracking bound with separate contributions from representation and head drift. This motivates comparing output-only, hidden-only, and full gating across learning settings. Output-only gating achieves higher mean accuracy than full gating on the three evaluated label-permutation benchmarks, while hidden-only gating achieves higher mean return than output-only gating under Gridworld dynamics shifts.

Contributions. First, we analyze continuous-time utility-gated dynamics and relate the expected displacement-based utility rate to gated local curvature under the stated assumptions, including a correction for non-equilibrium probability currents (Theorem 3.2). This theoretical quantity is distinct from the raw utility implemented in UPGD-W, whose association with curvature-related diagnostics we examine empirically. Second, we establish a distributional tracking bound with separate representation and head drift contributions (Theorem 3.5, Corollary 3.6). This provides a theoretical motivation for examining relative block drift when comparing gating placements. Third, we evaluate these placements in supervised continual learning and reinforcement learning. Output-only gating achieves higher mean accuracy than full gating on the three evaluated label-permutation benchmarks while applying utility modulation to only 0.16%–2.5% of the parameters. All layers continue to update. The gating variants perform similarly on Input-Permuted MNIST. Under controlled Gridworld dynamics shifts, hidden-only gating improves on output-only gating. Supplementary single-task MuJoCo experiments provide additional comparisons of gating placement.

Further discussion of related work appears in Appendix C.

2 Background

We study online continual supervised learning from examples (x_t, y_t) drawn from changing distributions $\mathcal{D}_t$. A neural network $f(\cdot;\theta)$, with $\theta \in \mathbb{R}^N$, is updated sequentially to minimize cumulative expected loss $\sum_t \mathbb{E}_{(x,y)\sim\mathcal{D}_t}[\ell(f(x;\theta_t), y)]$. Tasks change periodically through input or label permutations (Elsayed & Mahmood, 2024). Plasticity describes the capacity to learn from new data through continued optimization (Lyle et al., 2023). We assess it through current-task performance under a fixed training budget, without directly evaluating retention on earlier tasks. Reinforcement-learning experiments appear in Section 5.

Utility-Gated Perturbed Gradient Descent. UPGD-W (Elsayed & Mahmood, 2024) estimates parameter utility using $\tilde{u}_{t,i} = -g_{t,i}\theta_{t,i}$, where $g_{t,i}$ is the stochastic gradient. Exponential averaging, bias correction, global normalization, and a sigmoid produce the utility score $s_{t,i}$. With $a_{t,i} = 1 - s_{t,i}$, the update is

$$\theta_{t+1,i} = (1 - \eta\lambda)\theta_{t,i} - \eta a_{t,i}(g_{t,i} + \varepsilon_{t,i}), \quad \varepsilon_{t,i} \sim \mathcal{N}(0, \sigma^2). \tag{1}$$

Here, η is the learning rate, λ controls weight decay, and σ controls injected noise. Higher utility reduces both gradient and noise updates, while weight decay is applied separately. Setting $a_{t,i} = 1$ gives the continual Shrink and Perturb baseline (Ash & Adams, 2020; Elsayed & Mahmood, 2024). Utility tracking requires $\mathcal{O}(N)$ operations per update. Appendix A gives the normalization details, and Appendix I describes the moment-normalized MuJoCo implementation.

Layer-Selective Variants. We compare full gating, which applies utility modulation throughout the network, output-only gating, which applies it to the final linear layer, and hidden-only gating, which applies it to the hidden layers. Nongated layers use fixed multipliers on their gradient and noise updates. All layers continue to update in every variant. These comparisons examine where utility modulation is beneficial under different adaptation demands.

3 Theoretical Framework

We formulate continual learning as distributional tracking of a moving target for a decomposed model with feature representation ϕ and linear head W. Let $\theta = (\phi, W)$. The task-optimal parame-

ters are

$$\theta_t^* = (\phi_t^*, W_t^*) \in \arg\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}_t}[\ell(f(x;\theta), y)]. \tag{2}$$

3.1 Continuous-Time Dynamics

We analyze idealized continuous-time model of a smooth, parameter dependent gating and evaluate UPGD-W experimentally. The implemented gate additionally depends on running utility estimates, so the model captures the shared update structure without explicitly modeling the evolution of those estimates. To decouple the environmental shift from the localized optimization process, we implement a two-timescale framework. The index t denotes the slow timescale corresponding to the sequence of stationary tasks, whereas τ denotes the fast timescale governing the inner-task optimization. Under this regime, the environmental optimum θ_t^* remains fixed for the duration of the inner-task optimization. With the positive diagonal gating operator $\mathbf{A}_t(\theta) = \mathrm{diag}(a_{t,1}(\theta), \ldots, a_{t,N}(\theta))$, the generic inner-task update is

$$\theta(\tau+\eta;t) = \theta(\tau;t) - \eta\lambda\theta(\tau;t) - \eta\mathbf{A}_t(\theta(\tau;t))g(\tau;t) - \eta\mathbf{A}_t(\theta(\tau;t))\varepsilon(\tau;t). \tag{3}$$

Before proceeding, we establish the regularity conditions governing the localized empirical risk functional.

Assumption 3.1 (Regularity Conditions)**.** We impose the following assumptions.

- The diagonal gating operator $\mathbf{A}_t(\theta) = \mathrm{diag}(a_{t,1}(\theta), \ldots, a_{t,N}(\theta))$ has multipliers satisfying $0 < a_{t,i}(\theta) \leq 1$. It is continuously differentiable and, along with its divergence $\nabla_\theta \cdot \mathbf{A}^2(\theta)$, exhibits at most polynomial growth as $\|\theta\| \to \infty$.[1]
- For any t, the empirical risk functional $\mathcal{L}(\theta;t) = \mathbb{E}_{(x,y)\sim\mathcal{D}_t}[\ell(f(x;\theta),y)]$ is twice continuously differentiable and bounded from below. The gradient $\nabla_\theta\mathcal{L}(\theta;t)$ is L-Lipschitz continuous.[2]
- The regularized potential $V(\theta;t) = \frac{\lambda}{2}\|\theta\|^2 + \mathcal{L}(\theta;t)$ is coercive, i.e. it satisfies $\lim_{\|\theta\|\to\infty} V(\theta;t) = \infty$.

To extract the continuous-time limiting dynamics, we evaluate the limit as $\eta \to 0$. To prevent the stochastic contribution from collapsing into a degenerate deterministic flow, the variance of the discrete noise sequence $\varepsilon(\tau;t)$ must scale as $\mathcal{O}(\eta^{-1})$. By scaling the injection such that $\varepsilon(\tau;t) \sim \mathcal{N}(0, \sigma^2/\eta\mathbf{I})$, the cumulative discrete noise process weakly converges to an Itô stochastic integral with diffusion matrix $\sigma\mathbf{A}_t(\theta)$, whose infinitesimal form is $\sigma\mathbf{A}_t(\theta)dW(\tau;t)$, where $W(\tau;t)$ is a standard N-dimensional Wiener process (Karatzas & Shreve, 1991, Chapter 2.4; Oksendal, 2013, Chapter 3.1; Kloeden & Platen, 1992, Chapter 14). Suppressing the task index t to isolate the inner-task dynamics, the parameter trajectory is governed by the following Itô stochastic differential equation—an overdamped Langevin diffusion with multiplicative noise (Pavliotis, 2014, Chapter 4)

$$d\theta_\tau = b(\theta_\tau)d\tau + \sigma\mathbf{A}(\theta_\tau)dW_\tau, \tag{4}$$

where the drift vector field is defined as $b(\theta) = -(\lambda\theta + \mathbf{A}(\theta)\nabla_\theta\mathcal{L}(\theta))$.

For the stationary utility analysis, we assume that, for each fixed task t, the SDE in equation 4 admits a stationary density $\pi(\theta)$. The expectations and integrals used in the utility–curvature theorem are finite. The boundary contributions produced by integration by parts tend to zero as the integration domain expands to all of parameter space.

Let $\pi(\theta)$ denote this stationary density. In general, it is non-equilibrium and has an associated probability current $J(\theta)$ satisfying $\nabla_\theta \cdot J = 0$ (Bogachev et al., 2015, Chapters 2 and 4). For the multiplicative Itô diffusion, this current is

$$J(\theta) = \pi(\theta)b(\theta) - \frac{\sigma^2}{2}\nabla_\theta \cdot (\mathbf{A}^2(\theta)\pi(\theta)). \tag{5}$$

The continuous flow need not satisfy detailed balance. Stationarity requires the probability current to be divergence-free, $\nabla_\theta \cdot J(\theta) = 0$, but does not require $J(\theta) = \mathbf{0}$. For state-dependent anisotropic

[1] For a vector field F, $\nabla_\theta \cdot F = \sum_i \partial_{\theta_i} F_i$. For a matrix field D, we use the row-wise divergence $(\nabla_\theta \cdot D)_i = \sum_j \partial_{\theta_j} D_{ij}$. In particular, $[\nabla_\theta \cdot \mathbf{A}^2]_i = \partial_{\theta_i}(a_i^2)$ for diagonal $\mathbf{A}$.

[2] An operator $F : \mathbb{R}^N \to \mathbb{R}^N$ is L-Lipschitz continuous if $\exists\ L > 0$ such that $\|F(\theta) - F(\theta')\| \leq L\|\theta - \theta'\|$.

gates, $J(\theta) \neq \mathbf{0}$ is the generic non-equilibrium case. In contrast, when the noise is isotropic, i.e. $\mathbf{A}(\theta) = \mathbf{I}$ so that no gating is applied, detailed balance holds and $J(\theta) = \mathbf{0}$. The possible nonzero stationary current is central to the subsequent analysis. It quantifies the departure from the equilibrium curvature relation, and Theorem 3.2 shows that it appears as an explicit correction to the gated-curvature term in expected utility.

For the curvature analysis, we use displacement-based utility, distinct from UPGD-W's implemented raw utility $\tilde{u}_{t,i} = -g_{t,i}\theta_{t,i}$. Let $\Delta\theta_\tau = \theta(\tau + \Delta\tau; t) - \theta(\tau; t)$, with $\Delta\tau = \eta$. Its stationary expected rate is defined below, with the construction detailed in Appendix B.1.

$$\mathcal{U}_\infty \coloneqq \lim_{\Delta\tau \downarrow 0} \frac{1}{\Delta\tau} \mathbb{E}_\pi \left[-\nabla_\theta \mathcal{L}(\theta_\tau) \Delta\theta_\tau^\top \right]. \tag{6}$$

The expected coordinate-wise utility is $\mathbb{E}_\pi[u_i^\Delta] \coloneqq (\mathcal{U}_\infty)_{ii}$.

The proof uses the drift-correlation identity in Lemma B.1, given in Appendix B.1.

Theorem 3.2 (Utility as Spectral Filtration of Expected Curvature)**.** *Let the fast-timescale optimization be governed by the stochastic differential equation in equation 4 under Assumption 3.1 and the stationary utility conditions stated after equation 4. It holds that*

$$\mathcal{U}_\infty = \frac{\sigma^2}{2} \mathbb{E}_\pi[\nabla^2_{\theta\theta} \mathcal{L}(\theta) \mathbf{A}^2(\theta)] - \int_{\mathbb{R}^N} \nabla_\theta \mathcal{L}(\theta) J(\theta)^\top \, \mathrm{d}\theta, \tag{7}$$

$$\mathbb{E}_\pi[u_i^\Delta] = \frac{\sigma^2}{2} \mathbb{E}_\pi[\nabla^2_{ii} \mathcal{L}(\theta) a_i^2(\theta)] - \int_{\mathbb{R}^N} \partial_i \mathcal{L}(\theta) J_i(\theta) \, \mathrm{d}\theta, \tag{8}$$

$$\mathrm{Tr}(\mathcal{U}_\infty) = \frac{\sigma^2}{2} \mathbb{E}_\pi[\mathrm{Tr}(\nabla^2_{\theta\theta} \mathcal{L}(\theta) \mathbf{A}^2(\theta))], \tag{9}$$

$$\left| \mathbb{E}_\pi[u_i^\Delta] - \frac{\sigma^2}{2} \mathbb{E}_\pi[\nabla^2_{ii} \mathcal{L}(\theta) a_i^2(\theta)] \right| \leq \sqrt{\mathcal{G}_i \Sigma_i}, \tag{10}$$

where $\pi(\theta)$ is the stationary measure satisfying $\partial_\tau \pi = 0$, $\mathcal{G}_i = \mathbb{E}_\pi[(\partial_i \mathcal{L}(\theta))^2]$ is the coordinate-wise uncentered second moment of the empirical gradient, and $\Sigma_i = \mathbb{E}_\pi[(J_i(\theta)/\pi(\theta))^2]$.

- *When $\mathbf{A}(\theta) = \mathbf{I}$ (i.e. no gating is used and UPGD-W reduces to Shrink and Perturb in equation 14), the diffusion generated by the stochastic differential equation satisfies detailed balance, i.e., $J(\theta) = \mathbf{0}$. The stationary measure $\pi(\theta)$ reduces to the Gibbs-Boltzmann distribution $\pi(\theta) \propto \exp(-2\sigma^{-2} V(\theta))$ for the regularized potential $V(\theta) = \frac{\lambda}{2}\|\theta\|^2 + \mathcal{L}(\theta)$, and $\mathcal{U}_\infty = \frac{\sigma^2}{2} \mathbb{E}_\pi[\nabla^2_{\theta\theta} \mathcal{L}(\theta)]$, implying $\mathbb{E}_\pi[u_i^\Delta] = \frac{\sigma^2}{2} \mathbb{E}_\pi[\nabla^2_{ii} \mathcal{L}(\theta)]$..*
- *Additionally, if $\mathbf{A}(\theta) = \mathbf{I}$ and $\mathcal{L}(\theta; t)$ has a constant Hessian $\mathbf{H}$, then $\mathcal{U}_\infty = \frac{\sigma^2}{2}\mathbf{H}$, implying $\mathbb{E}_\pi[u_i^\Delta] = \frac{\sigma^2}{2} H_{ii}$.*

Theorem 3.2 relates expected displacement-based utility to local coordinate curvature and a non-equilibrium current correction. The center of the coordinate-wise bound is the gated curvature $\frac{\sigma^2}{2} \mathbb{E}_\pi[\nabla^2_{ii} \mathcal{L}(\theta) a_i^2(\theta)]$, while its half-width $\sqrt{\mathcal{G}_i \Sigma_i}$ bounds the contribution of the coordinate-wise probability current. When no gating is used, detailed balance holds, the current vanishes, and displacement-based utility reduces exactly to the noise-scaled stationary average of the diagonal Hessian. Under active gating, the current need not vanish; Σ_i quantifies the resulting non-equilibrium correction. Appendix E.2 reports empirical correlations between UPGD-W's implemented raw utility and curvature-related diagnostics across supervised, gridworld, and MuJoCo settings; these measurements do not directly test the theorem's displacement-based utility identity. The identity alone does not establish an optimal gating rule; we next examine how representation and head drift enter a distributional tracking bound.

Takeaway 1. Our analysis shows expected displacement-based utility isolates gated local curvature, subject to an explicit correction for non-equilibrium probability currents induced by the gating mechanism itself (Theorem 3.2). This formalizes how utility gating interacts with the localized geometry of the loss landscape.

3.2 Distributional Tracking under Environmental Drift

The fast-timescale analysis of Theorem 3.2 relates expected displacement-based utility to gated local curvature, but local curvature alone is insufficient to derive an optimal plasticity allocation in a continuously shifting environment. We therefore conduct a slow-timescale continuous-time distributional tracking analysis. We model the task index t as continuous time, so that the empirical risk $\mathcal{L}(\theta; t)$, state-dependent diagonal gating matrix $\mathbf{A}(\theta; t) \succ 0$, and corresponding target invariant measure π_t evolve over time. Let $\mu(t)$ denote the active parameter distribution, defined as the law of the following time-inhomogeneous diffusion:

$$d\theta(t) = -\big(\lambda\theta(t) + \mathbf{A}(\theta(t); t)\nabla_\theta \mathcal{L}(\theta(t); t)\big)dt + \sigma\mathbf{A}(\theta(t); t)dW(t),$$

We assume a quasi-stationary regime in which $\mu(t)$ approximately tracks the moving target measure π_t, defined as the invariant measure of the fast-timescale diffusion equation 4 with the gating operator and empirical risk frozen at time t.

The corresponding Fokker–Planck equation is given in Appendix B.2. The target distribution changes with both the objective and the gating matrix, and its evolution satisfies the continuity equation below.

$$\partial_t \pi_t + \nabla_\theta \cdot (\pi_t v_t) = 0, \tag{11}$$

where $v_t \in L^2(\pi_t)$ is the unique optimal transport vector field minimizing the kinetic energy $\mathbb{E}_{\pi_t}[\|v_t\|^2]$ (Villani, 2009, Chapter 8). We also need the following assumption to control the variation of tasks.

Assumption 3.3 (Temporal Regularity of the Environment)**.** The continuous environmental distribution shift parameterized by time $t \in [0, T]$ induces a smooth temporal variation in the objective functional. Specifically, the unregularized empirical loss $\mathcal{L}(\theta; t)$ is continuously differentiable with respect to t, and its temporal derivative $\partial_t \mathcal{L}(\theta; t)$ exhibits at most polynomial spatial growth as $\|\theta\| \to \infty$.

Assumption 3.4 (Measure and Transport Regularity)**.** For every $t \in [0, T]$, the active and target measures $\mu(t)$ and π_t admit sufficiently smooth positive densities with $\mu(t) \ll \pi_t$, and $\mathrm{KL}(\mu(0) \parallel \pi_0) < \infty$. The curve $t \mapsto \pi_t$ is absolutely continuous in the 2-Wasserstein space and admits a minimal velocity field v_t satisfying the continuity equation.

The gating matrix is uniformly positive definite: there exist constants $0 < a_{\min} \le a_{\max} < \infty$ such that $a_{\min}\mathbf{I} \preceq \mathbf{A}(\theta; t) \preceq a_{\max}\mathbf{I}$ for all (θ, t). The densities, probability currents, and derivatives appearing below have sufficient integrability and boundary decay to justify differentiation under the integral sign and integration by parts.

We quantify tracking error between the active marginal measure $\mu(t)$ and the moving target measure π_t using the dynamically preconditioned relative Fisher information, and state the following result (see Appendix B.2 for the proof).

Theorem 3.5 (Distributional Tracking Bound)**.** *Let Assumptions 3.1, 3.3, and 3.4 hold continuously for all $t \in [0, T]$, and assume the initial active measure $\mu(0)$ possesses finite exponential moments. Define the dynamically preconditioned relative Fisher information and cumulative geometric drift transport cost by*[3]

$$\mathcal{I}_{\mathbf{A}}(\mu \parallel \pi_t) := \mathbb{E}_\mu[\|\nabla_\theta \ln(\mu(t)/\pi_t)\|^2_{\mathbf{A}^2(\theta;t)}],$$
$$P_T(\mathbf{A}) := \int_0^T \mathbb{E}_{\mu(t)}[\|v_t\|^2_{\mathbf{A}^{-2}(\theta;t)}]\,\mathrm{d}t. \tag{12}$$

Assume $P_T(\mathbf{A}) < \infty$. Then the cumulative distributional tracking error satisfies the following bound.

$$\int_0^T \mathcal{I}_{\mathbf{A}}(\mu(t) \parallel \pi_t)\mathrm{d}t \le \frac{4}{\sigma^2}\,\mathrm{KL}(\mu(0) \parallel \pi_0) + \frac{4}{\sigma^4}P_T(\mathbf{A}). \tag{13}$$

The quantity bounded above is integrated relative Fisher information. Relating it to cumulative prediction-loss regret requires additional assumptions.

For a more interpretable result, we state the following corollary (proof in Appendix B.3).

[3] $\|x\|_{\mathbf{B}} := \sqrt{\langle x, \mathbf{B}x\rangle}$ for a positive definite matrix $\mathbf{B}$.

Corollary 3.6 (Blockwise Distributional Tracking Bound)**.** *Under the assumptions of Theorem 3.5, let the parameter space be partitioned as $\theta = (\phi, W)$ and assume the state-dependent diagonal gating matrix takes the block form $\mathbf{A}(\theta; t) = \mathrm{diag}(\alpha_\phi(\phi; t)\mathbf{I}_\phi, \alpha_W(W; t)\mathbf{I}_W)$ for positive scalar functions α_ϕ, α_W. Let the continuity equation of the target measure π_t be governed by the decoupled minimal velocity field $v_t = (v_t^\phi, v_t^W)$. Upon defining the localized instantaneous transport costs as $c_t^\phi(\theta) := \|v_t^\phi(\theta)\|^2$ and $c_t^W(\theta) := \|v_t^W(\theta)\|^2$, the cumulative distributional tracking error satisfies the following bound.*

$$\int_0^T \mathcal{I}_{\mathbf{A}}(\mu(t) \parallel \pi_t)\mathrm{d}t \leq \frac{4}{\sigma^2}\,\mathrm{KL}(\mu(0) \parallel \pi_0) + \frac{4}{\sigma^4}\int_0^T \left(\mathbb{E}_{\mu(t)}\left[\frac{c_t^\phi(\theta)}{\alpha_\phi^2(\phi; t)}\right] + \mathbb{E}_{\mu(t)}\left[\frac{c_t^W(\theta)}{\alpha_W^2(W; t)}\right]\right)\mathrm{d}t.$$

The blockwise bound separates representation and head transport costs, each weighted by the inverse squared gain of its block. This motivates comparing utility gating restricted to the output head with gating that also includes the representation. The relative performance of these placements is examined empirically across different adaptation settings.

To describe relative block movement empirically, we measure the normalized checkpoint-displacement ratio $\rho = (D_\phi/\sqrt{N_\phi})/(D_W/\sqrt{N_W})$ in ungated baseline runs. Here, D_l is the sum of checkpoint displacement magnitudes for block l, and N_l is its parameter count. The ratio is defined when the head path length is positive. A smaller ratio indicates movement concentrated more strongly in the head, while a larger ratio indicates greater representation movement relative to head movement. This ratio describes observed parameter movement rather than the target-distribution transport cost in the bound. We report this measurement alongside performance to interpret the gating comparisons, without imposing fixed thresholds for selecting a mode.

Takeaway 2. Our analysis shows the cumulative tracking error under continuous environmental drift can be bounded by decoupled costs for the representation and the output head (Corollary 3.6). This blockwise separation provides theoretical justification for layer-selective gating, motivating targeted interventions based on where adaptation demands occur.

We complement the theoretical analysis with a synthetic experiment that compares how allocating adaptation between two parameter blocks affects tracking performance (Appendix D). The following experiments compare output-only, hidden-only, and full gating in supervised continual learning and reinforcement learning.

4 Experiments

4.1 Supervised Continual Learning Setup

We evaluate four continual-learning benchmarks following Elsayed & Mahmood (2024). Three use label permutation: Mini-ImageNet (100 classes, ResNet-50 features) (Dohare et al., 2023), EMNIST (47 classes) (Javed & White, 2019), and CIFAR-10 (10 classes) (Krizhevsky, 2009). At each task boundary, the assignment of classes to output labels changes, while the input pixels or extracted features supplied for each example remain unchanged. Input-Permuted MNIST (LeCun et al., 2010) instead permutes input pixel coordinates while preserving class labels. These interventions define target shift and input shift, respectively; they do not constrain which network layers can adapt. Each run spans 10^6 steps with periodic task changes. We train a fully-connected ReLU network (input $\rightarrow$ 300 $\rightarrow$ 150 $\rightarrow$ output) online with batch size one (Appendix F.1).

We compare UPGD-W with SGD, Adam, Synaptic Intelligence, and Shrink & Perturb (Ash & Adams, 2020). The three UPGD-W variants apply utility gating to the output head only (output-only gating), hidden layers only (hidden-only gating), or all layers (full gating). All layers continue to update in every variant; nongated layers use fixed scaling. Our primary performance measure is final accuracy, averaged over the last 40 tasks. We additionally report short-horizon plasticity $(1 - \ell_{\text{after}}/\ell_{\text{before}})$ and dead-unit fraction as diagnostic measures in Appendix H.

4.2 Supervised Continual Learning Results

Accuracy and Layer-Selective Gating Figure 1 pairs learning trajectories with final accuracy across individual runs for each benchmark. The learning curves include all seven methods, while

the final-accuracy panels compare the three UPGD-W gating variants and Shrink & Perturb. Table 1 reports final accuracy for all methods.

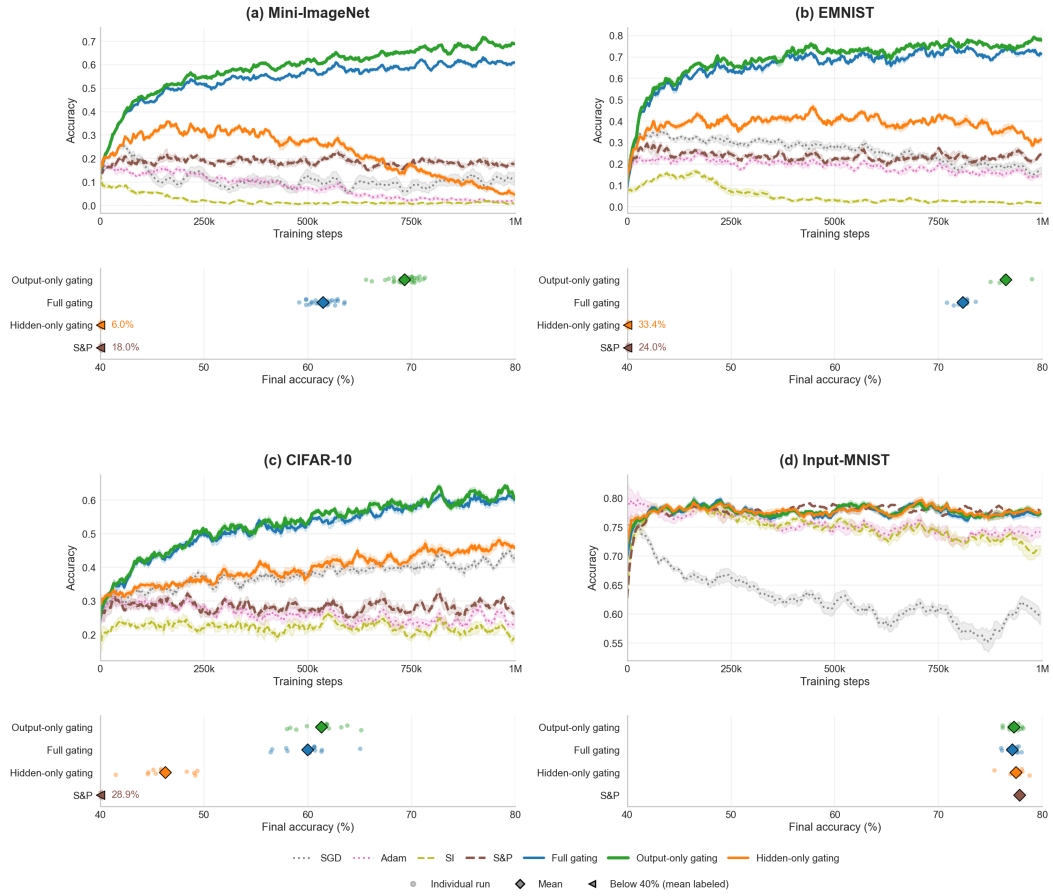


Figure 1: Learning trajectories and final accuracy on four continual-learning benchmarks. (a–c) Label-permuted Mini-ImageNet, EMNIST, and CIFAR-10. (d) Input-Permuted MNIST. Within each dataset, the upper panel shows per-task accuracy with a 10-task moving average and shading of ±1 SE across runs where available. The lower panel shows each run's mean accuracy over its last 40 tasks as a dot and the across-run mean as a diamond. Lower panels share a 40–80% accuracy scale. Left-pointing triangles mark values below 40%; outlined triangles mark means below 40%, with their actual values labeled. Input-MNIST S&P uses one run at the original learning rate of 0.001, without an SE band. Run counts appear in Appendix F.1.1.

Table 1: Final accuracy (mean ± SE where available). Bold indicates the highest mean on the label-permutation benchmarks; it does not imply statistical significance.

Method	Mini-ImageNet	EMNIST	CIFAR-10	Input-MNIST
SGD	0.106±0.018	0.176±0.015	0.423±0.005	0.586±0.005
Adam	0.022±0.002	0.154±0.005	0.251±0.006	0.740±0.003
SI	0.013±0.001	0.017±0.001	0.213±0.003	0.723±0.004
S&P	0.180±0.010	0.240±0.005	0.289±0.002	0.778±0.001
UPGD Full	0.615±0.002	0.724±0.003	0.606±0.000	0.775±0.002
UPGD Out	**0.693±0.003**	**0.765±0.004**	**0.618±0.000**	0.773±0.002
UPGD Hid	0.060±0.005	0.334±0.007	0.455±0.003	0.778±0.002

On the three evaluated label-permutation benchmarks, output-only UPGD-W has the highest reported mean final accuracy (Table 1). Its advantage over full gating is larger on Mini-ImageNet and EMNIST than on CIFAR-10. These results support output-only gating on the evaluated label-permutation benchmarks. The larger gains on datasets with more classes suggest a possible role for

output dimensionality, but these comparisons do not separate class count from other dataset differences.

On Input-Permuted MNIST, the reported mean accuracies of the three UPGD-W variants and S&P are close. Input permutation changes the features presented to the network, and adaptation can involve both the representation and its interaction with the readout. This coupling makes the relationship between shift type and beneficial gating location less direct than the simple input-versus-output distinction suggests. The observed results therefore do not indicate a strong preference for a particular gating configuration on this benchmark.

The label-permutation results reveal a consistent pattern across gating configurations. Full gating achieves higher mean final accuracy than hidden-only gating on all three benchmarks, supporting the benefit of adding utility modulation at the head. Output-only gating achieves still higher mean accuracy than full gating, indicating that extending modulation to the hidden layers does not improve performance in these experiments. Together, these comparisons support concentrating utility gating at the readout, where changing label assignments can be accommodated.

Output-only gating achieves these results while applying utility-dependent modulation to only 0.16%–2.5% of the network's parameters (Appendix G). All layers continue to update, so this small fraction describes the scope of gating, not the trainable portion of the network. The close results on Input-Permuted MNIST indicate that the preference for output-only gating depends on the learning setting. Extending utility gating throughout the network is therefore not necessarily beneficial: on the evaluated label-permutation tasks, restricting it to the output head yields higher mean accuracy. Additional utility-distribution measurements and clamping ablations examine the sensitivity of performance to utility modulation (Appendix F).

5 Reinforcement Learning Evaluation

We next examine utility-gating placement in continual reinforcement learning using Gridworld DQN under controlled reward, dynamics, and joint shifts. These interventions target adaptation to changing environment components, following the formulations reviewed by Khetarpal et al. (2022). Abel et al. (2023) distinguish continual reinforcement learning through the requirement that the best agents continue learning, relative to an agent basis. Our main evaluation therefore emphasizes repeated environmental changes rather than policy-induced changes in the training distribution alone. We also report four single-task MuJoCo PPO benchmarks in Appendix I. These supplementary experiments assess gating in standard reinforcement learning without imposed shifts.

The controlled Gridworld interventions specify which environment component changes, but adaptation can involve both the representation and the output head. We therefore measure normalized representation-to-head checkpoint displacement in ungated baseline runs and report the ratio ρ alongside the gating comparisons (Appendix E). Guided by the block tracking bound, we examine how relative block movement accompanies the performance of output-only, hidden-only, and full gating. The ratio describes the observed movement under each intervention. Gating placements are compared empirically, without assigning them through fixed ratio thresholds.

Gridworld provides evidence that gating hidden layers can improve performance beyond output-only gating under dynamics shifts. We evaluate DQN in a 7×7 environment with reward shifts induced by goal changes, dynamics shifts induced by obstacle-layout changes, and joint shifts combining both interventions. Figure 2 shows training over 50K episodes, and Table 2 reports post-shift plateau returns. Under dynamics shift, hidden-only gating achieves a higher mean return than output-only gating (0.555 ± 0.004 versus 0.507 ± 0.012). Output-only gating has the highest mean under reward shift, while full gating has the highest mean under joint shift. These latter comparisons do not establish a uniquely best gating variant. The dynamics-shift result shows a setting where gating hidden layers is beneficial.

The benefit of hidden-only gating over output-only gating under Gridworld dynamics shifts provides evidence that utility modulation in the representation can be useful when adapting to changes in the environment. This finding complements the advantage of output-only gating on the evaluated supervised label-permutation benchmarks. The block tracking bound motivates examining representation and head drift together, and the Gridworld comparisons show why gating placement beyond the output head deserves consideration.

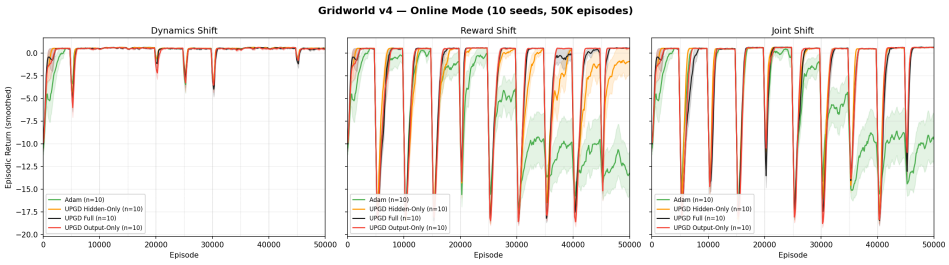


Figure 2: Gridworld DQN online training over 50K episodes under dynamics shift (left), reward shift (center), and joint shift (right). Curves show the across-seed mean (10 seeds) of training returns binned in 100-episode windows, smoothed with a 5-bin moving average that is not reset at phase boundaries; shaded bands show one standard error across seeds, smoothed identically. Vertical dotted lines mark phase boundaries.

Table 2: Gridworld DQN plateau returns over 50K training episodes, reported as mean ± SE across 10 runs. Within each run, returns are averaged over the 50 logged episode returns in the final 500 episodes of each of nine post-shift phases, then across phases. SE uses the sample SD across runs.

Method	Reward shift	Dynamics shift	Joint shift
Output-only gating	0.511 ± 0.006	0.507 ± 0.012	0.536 ± 0.009
Full gating	0.376 ± 0.100	0.530 ± 0.010	0.550 ± 0.004
Hidden-only gating	0.116 ± 0.315	0.555 ± 0.004	0.544 ± 0.004
Adam	-5.258 ± 1.286	0.556 ± 0.004	-3.567 ± 0.830

Table 3 relates the Gridworld gating comparisons to measured relative block movement. The ratio ρ increases from reward shift (0.490) to joint shift (0.530) to dynamics shift (0.570), indicating progressively greater representation movement relative to head movement. The corresponding highest mean returns occur with output-only, full, and hidden-only gating. This ordering is consistent with examining representation movement when interpreting the benefit of gating hidden layers, although it does not establish a ratio-based selection rule.

Table 3: Layer-selective gating comparisons across three controlled Gridworld shift conditions alongside measured representation-to-head displacement ratios. The highest mean return is descriptive and does not establish a uniquely best gating configuration. Measurement details for ρ are provided in Appendix E. Supplementary single-task MuJoCo comparisons appear in Table 22.

Environment	Measured ρ	Gating with highest mean return	Comparison notes
Gridworld (reward shift)	0.490	Output-only gating	Full and hidden-only gating show greater across-run variability.
Gridworld (dynamics shift)	0.570	Hidden-only gating	Higher mean return than output-only gating.
Gridworld (joint shift)	0.530	Full gating	Mean returns range from 0.536 to 0.550 across gating variants.

Appendix I provides the extended 100,000-episode Gridworld experiment and replay buffer ablation. Single-task MuJoCo PPO results appear in Appendix I.2 and Table 22. These benchmarks provide additional comparisons of gating placement without imposed reward or dynamics changes.

6 Conclusion

We studied how utility gating can be allocated across network layers in continual learning. Our analysis relates displacement-based utility to local curvature under the stated assumptions and establishes a tracking bound with separate contributions from representation and head drift. This provides a theoretical motivation for examining where adaptation occurs when comparing gating placements. Empirically, output-only gating achieves higher mean accuracy than full gating on the three evaluated label-permutation benchmarks while applying utility modulation to only 0.16%–2.5% of the parameters. The gating variants perform similarly on Input-Permuted MNIST. In Gridworld, hidden-only gating improves on output-only gating under controlled dynamics shifts, demonstrating that including hidden layers in utility modulation can benefit adaptation to environmental changes. Measured relative block movement provides complementary context for these findings. Together, the results support selective gating as a practical design choice and motivate comparing output-only and full gating according to the adaptation demands of the setting.

AI USE STATEMENT

Large language models supported selected aspects of this work, as detailed below.

Tasks requiring disclosure. We used large language models to review theoretical arguments and existing proofs, including their assumptions, consistency, and notation. Examples include clarifying the distinction between implemented parameter utility and the displacement-based utility used in the analysis, and checking notation across theorem statements and proofs. We also used these tools to review evaluation procedures and the interpretation of empirical results, including averaging Grid-world returns across phases.

Tasks with recommended disclosure. We used these tools to develop diagnostic and analysis scripts, check consistency between tables and captions, and review bibliography entries.

Author responsibility. The authors are responsible for validating the mathematical arguments, numerical analyses, and supporting code. We take responsibility for the final content, including all text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work uses standard public benchmarks and simulated control environments; it involves no human subjects, personal data, or foreseeable dual-use concerns beyond those of general continual-learning research.

REPRODUCIBILITY STATEMENT

The appendix describes the supervised experimental setup and run counts (Appendices F.1 and F.1.1), reinforcement learning architectures and training protocols (Appendix I), and the checkpoint-displacement measurement protocol (Appendix E). Code for the optimizer variants, environments, and analysis scripts will be released with the camera-ready version.

REFERENCES

Zaheer Abbas, Rosie Zhao, Joseph Modayil, Adam White, and Marlos C. Machado. Loss of plasticity in continual deep reinforcement learning. In *Conference on Lifelong Learning Agents*, 2023.

David Abel, André Barreto, Benjamin Van Roy, Doina Precup, Hado van Hasselt, and Satinder Singh. A definition of continual reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2307.11046.

Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *European Conference on Computer Vision*, pp. 139–154, 2018.

Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. *Gradient Flows in Metric Spaces and in the Space of Probability Measures*. Birkhäuser, 2nd edition, 2008. doi: 10.1007/978-3-7643-8722-8.

Sotiris Anagnostidis, Gregor Bachmann, Lorenzo Noci, and Thomas Hofmann. The curious case of benign memorization. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=4C8ChYvMYBn.

Jordan T. Ash and Ryan P. Adams. On warm-starting neural network training. In *Advances in Neural Information Processing Systems*, volume 33, pp. 3884–3894, 2020.

Krishna Balasubramanian, Sinho Chewi, Murat A Erdogdu, Adil Salim, and Shunshi Zhang. Towards a theory of non-log-concave sampling: first-order stationarity guarantees for langevin monte carlo. In *Conference on Learning Theory*, pp. 2896–2923. PMLR, 2022.

Tudor Berariu, Wojciech Czarnecki, Soham De, Jorg Bornschein, Samuel Smith, Razvan Pascanu, and Claudia Clopath. A study on the plasticity of neural networks. *CoRR*, abs/2106.00042, 2021.

Omar Besbes, Yonatan Gur, and Assaf Zeevi. Non-stationary stochastic optimization. *Operations Research*, 63(5):1227–1244, 2015. doi: 10.1287/opre.2015.1408.

Vladimir I. Bogachev, Nicolai V. Krylov, Michael Röckner, and Stanislav V. Shaposhnikov. *Fokker–Planck–Kolmogorov Equations*, volume 207 of *Mathematical Surveys and Monographs*. American Mathematical Society, 2015. doi: 10.1090/surv/207.

Shibhansh Dohare, Richard S. Sutton, and A. Rupam Mahmood. Continual backprop: Stochastic gradient descent with persistent randomness. *CoRR*, abs/2108.06325, 2021.

Shibhansh Dohare, J. Fernando Hernandez-Garcia, Parash Rahman, Richard S. Sutton, and A. Rupam Mahmood. Maintaining plasticity in deep continual learning. *CoRR*, abs/2306.13812, 2023.

Shibhansh Dohare, J. Fernando Hernandez-Garcia, Qingfeng Lan, Parash Rahman, A. Rupam Mahmood, and Richard S. Sutton. Loss of plasticity in deep continual learning. *Nature*, 2024. doi: 10.1038/s41586-024-07711-7.

Mohamed Elsayed and A. Rupam Mahmood. Addressing loss of plasticity and catastrophic forgetting in continual learning. In *International Conference on Learning Representations*, 2024.

Lawrence C Evans. *Partial differential equations*, volume 19. American mathematical society, 2022.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2019.

Caglar Gulcehre, Srivatsan Srinivasan, Jakub Sygnowski, Georg Ostrovski, Mehrdad Farajtabar, Matthew Hoffman, Razvan Pascanu, and Arnaud Doucet. An empirical study of implicit regularization in deep offline RL. *Transactions on Machine Learning Research*, 2022.

Naicheng He, Kaicheng Guo, Arjun Prakash, Saket Tiwari, Ruo Yu Tao, Tyrone Serapio, Amy Greenwald, and George Konidaris. Spectral collapse drives loss of plasticity in deep continual learning. *arXiv preprint arXiv:2509.22335*, 2025. URL https://arxiv.org/abs/2509.22335v1.

J. Fernando Hernandez-Garcia, Shibhansh Dohare, Jun Luo, and Richard S. Sutton. Reinitializing weights vs units for maintaining plasticity in neural networks. In *Proceedings of The 4th Conference on Lifelong Learning Agents*, volume 330 of *Proceedings of Machine Learning Research*, pp. 776–806. PMLR, 2026. URL https://proceedings.mlr.press/v330/hernandez-garcia26a.html.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning*, pp. 2790–2799. PMLR, 2019.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Khurram Javed and Martha White. Meta-learning representations for continual learning. In *Advances in Neural Information Processing Systems*, 2019.

Arthur Juliani and Jordan T. Ash. A study of plasticity loss in on-policy deep reinforcement learning. In *Advances in Neural Information Processing Systems*, 2024.

Ioannis Karatzas and Steven E. Shreve. *Brownian Motion and Stochastic Calculus*, volume 113 of *Graduate Texts in Mathematics*. Springer, 2nd edition, 1991. doi: 10.1007/978-1-4612-0949-2.

Khimya Khetarpal, Matthew Riemer, Irina Rish, and Doina Precup. Towards continual reinforcement learning: A review and perspectives. *Journal of Artificial Intelligence Research*, 2022. URL https://arxiv.org/abs/2012.13490.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

Timo Klein, Lukas Miklautz, Kevin Sidak, Claudia Plant, and Sebastian Tschiatschek. Plasticity loss in deep reinforcement learning: A survey. arXiv preprint arXiv:2411.04832, 2024.

Peter E. Kloeden and Eckhard Platen. *Numerical Solution of Stochastic Differential Equations*, volume 23. Springer Berlin, Heidelberg, 1992. doi: 10.1007/978-3-662-12616-5.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Aviral Kumar, Rishabh Agarwal, Dibya Ghosh, and Sergey Levine. Implicit under-parameterization inhibits data-efficient deep reinforcement learning. In *International Conference on Learning Representations*, 2021.

Saurabh Kumar, Henrik Marklund, and Benjamin Van Roy. Maintaining plasticity via regenerative regularization. *CoRR*, abs/2308.11958, 2023.

Yann LeCun, Corinna Cortes, and Christopher J.C. Burges. MNIST handwritten digit database, 2010. Available: http://yann.lecun.com/exdb/mnist.

Alex Lewandowski, Haruto Tanaka, Dale Schuurmans, and Marlos C. Machado. Directions of curvature as an explanation for loss of plasticity. *arXiv preprint arXiv:2312.00246*, 2023. URL https://arxiv.org/abs/2312.00246.

Alex Lewandowski, Michał Bortkiewicz, Saurabh Kumar, András György, Dale Schuurmans, Mateusz Ostaszewski, and Marlos C. Machado. Learning continually by spectral regularization. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/5565ab682d6c7f8d9da34ba0919974b0-Abstract-Conference.html.

Clare Lyle, Mark Rowland, and Will Dabney. Understanding and preventing capacity loss in reinforcement learning. In *International Conference on Learning Representations*, 2022.

Clare Lyle, Zeyu Zheng, Evgenii Nikishin, Bernardo Avila Pires, Razvan Pascanu, and Will Dabney. Understanding plasticity in neural networks. In *International Conference on Machine Learning*, pp. 23190–23211. PMLR, 2023.

Evgenii Nikishin, Max Schwarzer, Pierluca D'Oro, Pierre-Luc Bacon, and Aaron Courville. The primacy bias in deep reinforcement learning. *CoRR*, abs/2205.07802, 2022.

Bernt Oksendal. *Stochastic differential equations: an introduction with applications*. Springer Science & Business Media, 2013.

Grigorios A. Pavliotis. *Stochastic Processes and Applications: Diffusion Processes, the Fokker–Planck and Langevin Equations*, volume 60 of *Texts in Applied Mathematics*. Springer New York, 2014. doi: 10.1007/978-1-4939-1323-7.

Mark B. Ring. *Continual Learning in Reinforcement Environments*. The University of Texas at Austin, 1994.

Ghada Sokar, Rishabh Agarwal, Pablo Samuel Castro, and Utku Evci. The dormant neuron phenomenon in deep reinforcement learning. In *International Conference on Machine Learning*. PMLR, 2023.

Hongyao Tang, Johan Obando-Ceron, Pablo Samuel Castro, Aaron Courville, and Glen Berseth. Mitigating plasticity loss in continual reinforcement learning by reducing churn. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 58883–58904. PMLR, 2025. URL https://proceedings.mlr.press/v267/tang25g.html.

Sebastian Thrun. Lifelong learning algorithms. In *Learning to Learn*, pp. 181–209. Springer, 1998.

Cédric Villani. *Optimal Transport: Old and New*, volume 338 of *Grundlehren der mathematischen Wissenschaften*. Springer, 2009. doi: 10.1007/978-3-540-71050-9.

Jiuqi Wang, Jayanth Srinivasa, Claire Chen, Shuze Daniel Liu, Ali Payani, and Shangtong Zhang. Predicting plasticity in deep continual learning: A theoretical perspective, 2026. URL https://arxiv.org/abs/2605.09044.

Jiaheng Wei, Yanjun Zhang, Leo Yu Zhang, Ming Ding, Chao Chen, Kok-Leong Ong, Jun Zhang, and Yang Xiang. Memorization in deep learning: A survey. *arXiv preprint arXiv:2406.03880*, 2024. URL https://arxiv.org/abs/2406.03880.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems*, volume 27, pp. 3320–3328, 2014.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*, pp. 3987–3995. PMLR, 2017.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning*, pp. 928–936. AAAI Press, 2003.

APPENDIX TABLE OF CONTENTS

A Notation and Setup **15**

B Full Proofs **15**

B.1 Proof of Theorem 3.2 15

B.2 Proof of Theorem 3.5 19

B.3 Proof of Corollary 3.6 21

B.4 Conditional Allocation under a Fixed Gain Budget 21

B.5 Scope of the Regularity Assumptions 22

C Extended Related Work **22**

C.1 Plasticity loss and its measurement 22

C.2 Methods for maintaining plasticity and reducing forgetting 23

C.3 Utility gating and layer-selective adaptation 23

C.4 Connections to tracking and curvature 23

D Synthetic Tracking Unit Test **23**

E Relative Block Movement and Utility–Curvature Diagnostics **25**

E.1 Relative Block Movement from Training Checkpoints 25

E.2 Empirical Utility-Curvature Diagnostics 26

F Supervised Experimental Setup and Utility Analysis **27**

F.1 Experimental Setup 27

F.1.1 Run Records for Final Accuracy 27

F.2 Per-Layer Utility Statistics 27

F.3 Utility Distributions and Gate Selectivity 27

F.4 Accuracy under Utility Clamping 28

G Gating Coverage and Computational Cost **28**

H Diagnostic Metrics for Dead Units and Plasticity **29**

I Reinforcement Learning Evaluation Details **30**

I.1 Gridworld 30

I.2 MuJoCo 34

J Scope and Limitations **37**

A Notation and Setup

Consider a neural network decomposed as representation plus head

$$f(x;\theta) = Wh(x;\phi),$$

Here $\phi \in \mathbb{R}^{d_\phi}$ denotes the representation parameters, $h(x;\phi)$ is the learned feature vector, and W is the linear head with d_W parameters. We identify each parameter block with its vectorization and write $\theta = (\phi, W) \in \mathbb{R}^N$, where $N = d_\phi + d_W$. The symbol $V(\theta;t)$ denotes the regularized potential, not a parameter block.

Definition A.1 (UPGD-W Update)**.** The UPGD-W update (Elsayed & Mahmood, 2024) for parameter θ_i at time t is formulated as

$$\theta_{t+1,i} = (1-\eta\lambda)\theta_{t,i} - \eta a_{t,i}(g_{t,i} + \varepsilon_{t,i}),$$

where $g_{t,i} = \partial\ell_t/\partial\theta_i$ is the empirical gradient, $\varepsilon_{t,i} \sim \mathcal{N}(0,\sigma^2)$ is the independent injected noise, and $\lambda \geq 0$ denotes the weight decay coefficient.

Definition A.2 (Utility and Gating)**.** Let $m_{0,i} = 0$ and $\beta \in [0,1)$. The raw utility is defined as $\tilde{u}_{t,i} = -g_{t,i} \cdot \theta_{t,i}$. The bias-corrected utility utilizes an exponential moving average

$$\hat{u}_{t,i} = \frac{m_{t,i}}{1-\beta^t}, \quad m_{t,i} = \beta m_{t-1,i} + (1-\beta)\tilde{u}_{t,i}.$$

The gate is constructed via global-maximum normalization followed by a sigmoid non-linearity

$$s_{t,i} = \varsigma\left(\frac{\hat{u}_{t,i}}{d_t}\right), \quad d_t = \max\left(\max_{j\in[N]} m_{t,j},\, 10^{-8}\right),$$
$$a_{t,i} = 1 - s_{t,i},$$

The normalizer d_t uses the signed maximum of the uncorrected moving averages with a positive floor, whereas the numerator is bias-corrected. The bias correction approaches one as t increases. This discrete gate depends on running statistics. The continuous-time theory in the main text instead treats the gate as a smooth function of the current parameters and time.

Here, $\varsigma(x) = 1/(1+e^{-x})$ is the standard logistic sigmoid.

The continual Shrink and Perturb baseline (Ash & Adams, 2020; Elsayed & Mahmood, 2024) sets $a_{t,i} = 1$:

$$\theta_{t+1,i} = (1-\eta\lambda)\theta_{t,i} - \eta(g_{t,i} + \varepsilon_{t,i}). \tag{14}$$

B Full Proofs

B.1 Proof of Theorem 3.2

For the curvature analysis, we introduce a theoretical displacement-based utility, distinct from UPGD-W's implemented raw utility $\tilde{u}_{t,i} = -g_{t,i}\theta_{t,i}$. Under the diffusion approximation, we use the population gradient and define $\tilde{u}^{\Delta}_{t,i} = -[\nabla_\theta \mathcal{L}(\theta_\tau)]_i[\Delta\theta_\tau]_i$. In matrix form, $\tilde{\mathcal{U}}(\tau) = -\nabla_\theta \mathcal{L}(\theta_\tau)\Delta\theta_\tau^\top$, where $\Delta\theta_\tau = \theta(\tau+\eta;t) - \theta(\tau;t)$ and $\Delta\theta_\tau \to 0$ almost surely as $\eta \downarrow 0$.[4] Because the parameter displacement includes injected stochastic noise, $\tilde{\mathcal{U}}(\tau)$ is random. For a fixed slow-time task t, we assume that the fast-timescale dynamics have relaxed to their stationary distribution $\pi(\theta)$; averaging with respect to π yields a time-independent expected characterization for

[4]Here we switch from the discrete increments used in the algorithmic implementation to the continuous increments $\Delta\tau := \eta$ for the ensuing analysis, where $\Delta\theta_\tau = \theta(\tau+\eta;t) - \theta(\tau;t)$.

the task. Normalizing by $\Delta\tau := \eta$ and taking the limit as the time step approaches zero from above yields the stationary displacement-based utility rate operator defined in equation 6.

With $\mathbf{e}_i$ denoting the ith standard basis, the expected coordinate-wise utility is

$$\mathbb{E}_\pi[u_i^\Delta] := \mathbf{e}_i^\top \mathcal{U}_\infty \mathbf{e}_i = \lim_{\Delta\tau\downarrow 0} \frac{1}{\Delta\tau}\mathbb{E}_\pi\left[-[\nabla_\theta \mathcal{L}(\theta_\tau)]_i[\Delta\theta_\tau]_i\right]. \tag{15}$$

We first establish the auxiliary drift-correlation identity used in the proof.

Lemma B.1. *Consider equation 4 and equation 6. Under Assumption 3.1 and the stationary utility conditions stated after equation 4, it holds that*

$$\mathcal{U}_\infty = \mathbb{E}_\pi[-\nabla_\theta \mathcal{L}(\theta) b(\theta)^\top]. \tag{16}$$

Proof of Lemma B.1. Fix the task and suppress its index. Under Assumption 3.1, the drift $b(\theta) = -(\lambda\theta + \mathbf{A}(\theta)\nabla\mathcal{L}(\theta))$ and diffusion coefficient $\sigma\mathbf{A}(\theta)$ are locally Lipschitz. Bounded gates and the globally Lipschitz loss gradient give

$$\|b(\theta)\| \le (|\lambda| + L)\|\theta\| + \|\nabla\mathcal{L}(0)\|, \qquad \|\sigma\mathbf{A}(\theta)\|_F^2 \le \sigma^2 N.$$

Thus the coefficients have at most linear growth. For a deterministic initial state, or an almost surely finite initial state measurable at time zero with Brownian increments independent of the initial state, the standard local-Lipschitz and linear-growth result gives a unique non-explosive strong solution (Oksendal, 2013, Chapter 5). Moreover, for every finite $H > 0$,

$$\mathbb{E}\left[\int_0^H \|\mathbf{A}(\theta_s)\|_F^2 \,\mathrm{d}s\right] \le NH < \infty.$$

The stochastic integral therefore has the square integrability needed for the martingale-increment argument below. Existence of the stationary density and the integrability and boundary decay required for the stationary expectations remain separate assumptions.

Substituting the integral representation of equation 4

$$\theta_{\tau+\Delta\tau} - \theta_\tau = \int_\tau^{\tau+\Delta\tau} b(\theta_s)ds + \int_\tau^{\tau+\Delta\tau} \sigma\mathbf{A}(\theta_s)dW_s \tag{17}$$

directly into equation 6 yields

$$\mathcal{U}_\infty = \lim_{\Delta\tau\downarrow 0} \frac{1}{\Delta\tau}\mathbb{E}_\pi\left[-\nabla_\theta\mathcal{L}(\theta_\tau)\left(\int_\tau^{\tau+\Delta\tau} b(\theta_s)ds + \int_\tau^{\tau+\Delta\tau}\sigma\mathbf{A}(\theta_s)dW_s\right)^\top\right]. \tag{18}$$

By linearity of expectation, and since $\mathbf{A}(\theta_s)$ is diagonal (hence symmetric) so that $(\mathbf{A}(\theta_s)dW_s)^\top = dW_s^\top\mathbf{A}(\theta_s)$,

$$\mathcal{U}_\infty = \lim_{\Delta\tau\downarrow 0}\frac{1}{\Delta\tau}\left(\mathbb{E}_\pi\left[-\nabla_\theta\mathcal{L}(\theta_\tau)\int_\tau^{\tau+\Delta\tau} b(\theta_s)^\top ds\right] + \sigma\mathbb{E}_\pi\left[-\nabla_\theta\mathcal{L}(\theta_\tau)\int_\tau^{\tau+\Delta\tau} dW_s^\top\mathbf{A}(\theta_s)\right]\right). \tag{19}$$

Because θ_τ is $\mathcal{F}_\tau$-measurable, so is $\nabla_\theta\mathcal{L}(\theta_\tau)$. Moreover, $\mathbf{A}(\theta_s)$ is predictable and square-integrable on $[\tau, \tau+\Delta\tau]$ by the bounded-gate estimate above. Hence, the Itô integral is a martingale increment (Karatzas & Shreve, 1991, Chapter 3; Oksendal, 2013, Chapter 3), and

$$\mathbb{E}\left[\int_{\tau}^{\tau+\Delta\tau} dW_s^\top \mathbf{A}(\theta_s)\middle|\mathcal{F}_\tau\right] = \mathbf{0}. \tag{20}$$

Thus, by the tower property, the second term in equation 19 is zero.

Accordingly, continuity of the sample paths and of b implies that the time average of $b(\theta_s)$ converges almost surely to $b(\theta_\tau)$ as $\Delta\tau \downarrow 0$. The growth conditions in Assumption 3.1 and the integrability conditions stated after equation 4 imply uniform integrability, permitting this limit to pass through expectation:

$$\mathcal{U}_\infty = \lim_{\Delta\tau\downarrow 0} \mathbb{E}_\pi\left[-\nabla_\theta\mathcal{L}(\theta_\tau)\left(\frac{1}{\Delta\tau}\int_{\tau}^{\tau+\Delta\tau} b(\theta_s)^\top ds\right)\right] = \mathbb{E}_\pi\left[-\nabla_\theta\mathcal{L}(\theta_\tau)b(\theta_\tau)^\top\right]. \tag{21}$$

As the stationary measure π is time-invariant, we can suppress τ to obtain $\mathbb{E}_\pi[-\nabla_\theta\mathcal{L}(\theta)b(\theta)^\top]$, thereby finishing the proof.

□

Proof of Theorem 3.2. By Lemma B.1 we have

$$\mathcal{U}_\infty = \mathbb{E}_\pi\left[-\nabla_\theta\mathcal{L}(\theta)b(\theta)^\top\right]. \tag{22}$$

By the stationary Fokker-Planck equation for Itô diffusions with multiplicative noise, the divergence of the probability current $J(\theta)$ vanishes (Bogachev et al., 2015, Chapters 2 and 4), i.e.

$$\nabla_\theta \cdot J(\theta) = \nabla_\theta \cdot \left(\pi(\theta)b(\theta) - \frac{\sigma^2}{2}\nabla_\theta \cdot (\mathbf{A}^2(\theta)\pi(\theta))\right) = 0. \tag{23}$$

Using the definition of divergence operator and rearrangement

$$\pi(\theta)b(\theta) = J(\theta) + \frac{\sigma^2}{2}\nabla_\theta \cdot (\mathbf{A}^2(\theta)\pi(\theta)). \tag{24}$$

Substituting this into equation 22 yields

$$\mathcal{U}_\infty = -\int_{\mathbb{R}^N} \nabla_\theta\mathcal{L}(\theta)J(\theta)^\top d\theta - \frac{\sigma^2}{2}\int_{\mathbb{R}^N} \nabla_\theta\mathcal{L}(\theta)\left(\nabla_\theta \cdot (\mathbf{A}^2(\theta)\pi(\theta))\right)^\top d\theta. \tag{25}$$

We evaluate the second integral via multidimensional integration by parts (Evans, 2022, Appendix C). Define $\mathbf{M}(\theta) = \mathbf{A}^2(\theta)\pi(\theta)$. Let $n(\theta)$ denote the outward unit normal to ∂B_R and dS the scalar surface measure. Applying the Divergence Theorem (Evans, 2022, Appendix C) over an expanding compact domain $B_R \subset \mathbb{R}^N$ of radius R yields

$$\begin{aligned}\int_{B_R} \nabla_\theta\mathcal{L}(\theta)\left(\nabla_\theta \cdot \mathbf{M}(\theta)\right)^\top d\theta = &\oint_{\partial B_R} \nabla_\theta\mathcal{L}(\theta)\big(\mathbf{M}(\theta)n(\theta)\big)^\top dS \\ &- \int_{B_R} \nabla^2_{\theta\theta}\mathcal{L}(\theta)\mathbf{M}(\theta)d\theta.\end{aligned} \tag{26}$$

By the explicit integrability and boundary conditions stated after equation 4, the boundary flux vanishes as $R \to \infty$, so that

$$\lim_{R\to\infty} \oint_{\partial B_R} \nabla_\theta\mathcal{L}(\theta)\big(\mathbf{M}(\theta)n(\theta)\big)^\top dS = \mathbf{0}. \tag{27}$$

Consequently, by extending the domain of integration to the entire parameter space we have

$$-\int_{\mathbb{R}^N} \nabla_\theta \mathcal{L}(\theta) \left(\nabla_\theta \cdot \mathbf{M}(\theta)\right)^\top d\theta = \int_{\mathbb{R}^N} \nabla^2_{\theta\theta} \mathcal{L}(\theta) \mathbf{M}(\theta) d\theta = \mathbb{E}_\pi[\nabla^2_{\theta\theta} \mathcal{L}(\theta) \mathbf{A}^2(\theta)]. \tag{28}$$

Substituting this expectation in equation 25 establishes the stated general identity in equation 7, from which we can readily obtain the coordinate-wise statement in equation 8 using the definition in equation 15.

Applying the matrix trace operator to our general identity yields

$$\mathrm{Tr}(\mathcal{U}_\infty) = \frac{\sigma^2}{2} \mathbb{E}_\pi[\mathrm{Tr}(\nabla^2_{\theta\theta} \mathcal{L}(\theta) \mathbf{A}^2(\theta))] - \int_{\mathbb{R}^N} \mathrm{Tr}(\nabla_\theta \mathcal{L}(\theta) J(\theta)^\top) d\theta. \tag{29}$$

By trace properties

$$\mathrm{Tr}(\nabla_\theta \mathcal{L}(\theta) J(\theta)^\top) = \nabla_\theta \mathcal{L}(\theta)^\top J(\theta) = \nabla_\theta \mathcal{L}(\theta) \cdot J(\theta). \tag{30}$$

We apply the product rule for the divergence operator to obtain

$$\nabla_\theta \cdot (\mathcal{L}(\theta) J(\theta)) = \nabla_\theta \mathcal{L}(\theta) \cdot J(\theta) + \mathcal{L}(\theta)(\nabla_\theta \cdot J(\theta)) = \nabla_\theta \mathcal{L}(\theta) \cdot J(\theta), \tag{31}$$

where we used $\nabla_\theta \cdot J(\theta) = 0$.

Integrating this identity over an expanding compact domain $B_R \subset \mathbb{R}^N$ and invoking the Divergence Theorem (Evans, 2022, Appendix C) provides

$$\int_{B_R} \nabla_\theta \mathcal{L}(\theta) \cdot J(\theta) d\theta = \oint_{\partial B_R} \mathcal{L}(\theta) J(\theta) \cdot d\mathbf{S}. \tag{32}$$

By the explicit integrability and boundary conditions stated after equation 4, the boundary surface integral above vanishes as $R \to \infty$, yielding the trace identity in equation 9.

To bound the deviation of the individual coordinates in equation 8, let us define

$$R = -\int_{\mathbb{R}^N} \nabla_\theta \mathcal{L}(\theta) J(\theta)^\top d\theta, \qquad R_{ii} = -\int_{\mathbb{R}^N} \partial_i \mathcal{L}(\theta) J_i(\theta) d\theta, \tag{33}$$

where $\partial_i \mathcal{L}(\theta)$ and $J_i(\theta)$ denote the i-th entry of their corresponding vectors. Upon doing so we can write

$$\begin{aligned} |R_{ii}| &= \left| \int_{\mathbb{R}^N} \partial_i \mathcal{L}(\theta) J_i(\theta) d\theta \right| \\ &= \left| \int_{\mathbb{R}^N} \left(\partial_i \mathcal{L}(\theta) \sqrt{\pi(\theta)} \right) \left(\frac{J_i(\theta)}{\sqrt{\pi(\theta)}} \right) d\theta \right| \\ &\leq \left(\int_{\mathbb{R}^N} (\partial_i \mathcal{L}(\theta))^2 \pi(\theta) d\theta \right)^{1/2} \left(\int_{\mathbb{R}^N} \frac{J_i(\theta)^2}{\pi(\theta)} d\theta \right)^{1/2}. \end{aligned} \tag{34}$$

by the Cauchy-Schwarz inequality. Denoting these factors with $\mathcal{G}_i$ and Σ_i, using the definition of expectation with respect to π, and plugging the results into the general coordinate-wise identity in equation 8 gives us the stated result in equation 10.

For the isotropic noise regime where $\mathbf{A}(\theta) = \mathbf{I}$, i.e. no gating, such that the drift vector field is $b(\theta) = -\nabla_\theta V(\theta)$, where $V(\theta) = \frac{\lambda}{2}\|\theta\|^2 + \mathcal{L}(\theta)$ is the regularized loss functional. The corresponding stationary Fokker-Planck equation admits the exact Gibbs measure (Pavliotis, 2014, Chapter 4)

$$\pi(\theta) = \frac{1}{Z} \exp\left(-\frac{2}{\sigma^2} V(\theta)\right), \tag{35}$$

for some normalization constant Z and consequently, not only the divergence of the probability current $J(\theta)$ vanishes, but also $J(\theta)$ itself vanishes such that

$$J(\theta) = \pi(\theta) b(\theta) - \frac{\sigma^2}{2} \nabla_\theta \pi(\theta) = \mathbf{0}, \tag{36}$$

Here we used $\nabla_\theta \pi(\theta) = (2/\sigma^2)\pi(\theta) b(\theta)$. The stationary probability current therefore vanishes, so the current-correction term in equation 7 is zero.

Finally, if the empirical risk $\mathcal{L}(\theta; t)$ has a constant Hessian $\mathbf{H}$, the expected Hessian $\mathbb{E}_\pi[\nabla^2_{\theta\theta} \mathcal{L}(\theta)]$ is readily $\mathbf{H}$ itself, concluding the proof.

□

B.2 Proof of Theorem 3.5

Let the continuous distribution shift be parameterized by a time-varying vector $\xi(t)$, such that the empirical risk is explicitly defined as $\mathcal{L}(\theta; \xi(t))$. Consequently, the target invariant measure is parametrically coupled to the joint trajectory of the environment and the gating matrix, denoted as $\pi_t \equiv \pi(\theta; \xi(t), \mathbf{A}(\theta; t))$. The temporal variation $\partial_t \pi_t$ is therefore driven by the environmental velocity $\dot{\xi}(t)$ and the macroscopic plasticity dynamics $\partial_t \mathbf{A}(\theta; t)$. The evolution of this moving target measure is described by the continuity equation stated in the main text.

Proof of Theorem 3.5. The proof extends the argument of (Balasubramanian et al., 2022, Lemma 23) to a moving target distribution and anisotropic noise induced by gating. Integration by parts yields the distributional tracking bound in the norm defined by the gating matrix.

Differentiating the KL divergence with respect to time under the integral sign yields (Ambrosio et al., 2008, Chapter 10)

$$\frac{d}{dt} \mathrm{KL}(\mu(t) \parallel \pi_t) = \int_{\mathbb{R}^N} \partial_t \mu(t) \ln \frac{\mu(t)}{\pi_t} \mathrm{d}\theta - \int_{\mathbb{R}^N} \mu(t) \partial_t \ln \pi_t \mathrm{d}\theta. \tag{37}$$

Substituting the Fokker–Planck equation governing the active measure (Bogachev et al., 2015, Chapter 2) gives

$$\partial_t \mu(t) = -\nabla_\theta \cdot (\mu(t) f_t) + \frac{\sigma^2}{2} \nabla_\theta \cdot (\nabla_\theta \cdot (\mathbf{A}^2(\theta; t) \mu(t))), \tag{38}$$

where the deterministic drift vector field is defined as $f_t = -(\lambda\theta + \mathbf{A}(\theta; t) \nabla_\theta \mathcal{L}(\theta; t))$. We express this drift via the stationary probability current $J_t = \pi_t f_t - \frac{\sigma^2}{2} \nabla_\theta \cdot (\mathbf{A}^2(\theta; t) \pi_t)$ of the target measure π_t. Isolating f_t and substituting it into $\tilde{J}_{\mu(t)} = \mu(t) f_t - \frac{\sigma^2}{2} \nabla_\theta \cdot (\mathbf{A}^2(\theta; t) \mu(t))$ provides

$$\tilde{J}_{\mu(t)} = \mu(t) \frac{J_t}{\pi_t} + \frac{\sigma^2}{2} \left[\frac{\mu(t)}{\pi_t} \nabla_\theta \cdot (\mathbf{A}^2(\theta; t) \pi_t) - \nabla_\theta \cdot (\mathbf{A}^2(\theta; t) \mu(t)) \right]. \tag{39}$$

For a diagonal operator $\mathbf{A}^2(\theta; t)$, the i-th vector component of the bracketed differential term simplifies to

$$\frac{\mu(t)}{\pi_t} \partial_i (A_i^2(\theta; t) \pi_t) - \partial_i (A_i^2(\theta; t) \mu(t)) = A_i^2(\theta; t) \left(\frac{\mu(t)}{\pi_t} \partial_i \pi_t - \partial_i \mu(t) \right) = -A_i^2(\theta; t) \mu(t) \partial_i \ln \frac{\mu(t)}{\pi_t}. \tag{40}$$

The derivatives of the gating operator $\mathbf{A}^2(\theta;t)$ cancel, giving

$$\tilde{J}_{\mu(t)} = \mu(t)\frac{J_t}{\pi_t} - \frac{\sigma^2}{2}\mu(t)\mathbf{A}^2(\theta;t)\nabla_\theta \ln \frac{\mu(t)}{\pi_t}. \tag{41}$$

By factoring the divergence operator, the Fokker-Planck equation constitutes the continuity equation $\partial_t \mu(t) = -\nabla_\theta \cdot \tilde{J}_{\mu(t)}$. Substituting this into the first integral of equation 37 and applying multidimensional integration by parts (Evans, 2022, Appendix C) yields

$$\begin{aligned}
\int_{\mathbb{R}^N} \left(-\nabla_\theta \cdot \tilde{J}_{\mu(t)}\right) \ln \frac{\mu(t)}{\pi_t} \mathrm{d}\theta &= \int_{\mathbb{R}^N} \left\langle \tilde{J}_{\mu(t)}, \nabla_\theta \ln \frac{\mu(t)}{\pi_t} \right\rangle \mathrm{d}\theta \\
&= \int_{\mathbb{R}^N} \left\langle \mu(t)\frac{J_t}{\pi_t} - \frac{\sigma^2}{2}\mu(t)\mathbf{A}^2(\theta;t)\nabla_\theta \ln \frac{\mu(t)}{\pi_t}, \nabla_\theta \ln \frac{\mu(t)}{\pi_t} \right\rangle \mathrm{d}\theta \\
&= -\frac{\sigma^2}{2}\mathcal{I}_{\mathbf{A}}(\mu(t) \parallel \pi_t) + \int_{\mathbb{R}^N} \left\langle \nabla_\theta \ln \frac{\mu(t)}{\pi_t}, J_t \right\rangle \frac{\mu(t)}{\pi_t} \mathrm{d}\theta \\
&= -\frac{\sigma^2}{2}\mathcal{I}_{\mathbf{A}}(\mu(t) \parallel \pi_t) + \int_{\mathbb{R}^N} \left\langle \nabla_\theta \left(\frac{\mu(t)}{\pi_t}\right), J_t \right\rangle \mathrm{d}\theta \\
&= -\frac{\sigma^2}{2}\mathcal{I}_{\mathbf{A}}(\mu(t) \parallel \pi_t) - \int_{\mathbb{R}^N} \frac{\mu(t)}{\pi_t}(\nabla_\theta \cdot J_t) \mathrm{d}\theta \\
&= -\frac{\sigma^2}{2}\mathcal{I}_{\mathbf{A}}(\mu(t) \parallel \pi_t).
\end{aligned} \tag{42}$$

The fourth equality follows from $\frac{\mu(t)}{\pi_t}\nabla_\theta \ln \frac{\mu(t)}{\pi_t} = \nabla_\theta(\frac{\mu(t)}{\pi_t})$. The fifth equality applies multidimensional integration by parts. Its boundary term vanishes by the integrability and boundary-decay conditions in Assumption 3.4. The final equality uses that J_t is the stationary current of π_t and therefore satisfies $\nabla_\theta \cdot J_t = 0$ (Bogachev et al., 2015, Chapter 4).

For the second integral in equation 37, we substitute the continuity equation $\partial_t \pi_t = -\nabla_\theta \cdot (\pi_t v_t)$ governing the optimal transport of the target measure, yielding

$$-\int_{\mathbb{R}^N} \mu(t)\partial_t \ln \pi_t \mathrm{d}\theta = \int_{\mathbb{R}^N} \frac{\mu(t)}{\pi_t}\nabla_\theta \cdot (\pi_t v_t)\mathrm{d}\theta. \tag{43}$$

Applying multidimensional integration by parts

$$\begin{aligned}
\int_{\mathbb{R}^N} \frac{\mu(t)}{\pi_t}\nabla_\theta \cdot (\pi_t v_t)\mathrm{d}\theta &= -\int_{\mathbb{R}^N} \left\langle \nabla_\theta \left(\frac{\mu(t)}{\pi_t}\right), \pi_t v_t \right\rangle \mathrm{d}\theta \\
&= -\int_{\mathbb{R}^N} \left\langle \frac{\pi_t \nabla_\theta \mu(t) - \mu(t)\nabla_\theta \pi_t}{\pi_t^2}, \pi_t v_t \right\rangle \mathrm{d}\theta \\
&= -\int_{\mathbb{R}^N} \left\langle \frac{\nabla_\theta \mu(t)}{\mu(t)} - \frac{\nabla_\theta \pi_t}{\pi_t}, v_t \right\rangle \mu(t)\mathrm{d}\theta \\
&= -\int_{\mathbb{R}^N} \left\langle \nabla_\theta \ln \frac{\mu(t)}{\pi_t}, v_t \right\rangle \mu(t)\mathrm{d}\theta.
\end{aligned} \tag{44}$$

The first equality follows by integration by parts, with $\oint_{\partial\mathbb{R}^N} \mu(t)v_t \cdot \mathrm{d}\mathbf{S} = 0$ under the integrability and boundary-decay conditions of Assumption 3.4. The remaining equalities apply the logarithmic derivative identity.

We apply Young's inequality for products $\langle a, b\rangle \leq \frac{1}{2\gamma}\|a\|^2 + \frac{\gamma}{2}\|b\|^2$ with $\gamma > 0$ to the standard Euclidean inner product by noting $\mathbf{I} = \mathbf{A}(\theta;t)\mathbf{A}^{-1}(\theta;t)$. Setting the scaling parameter to $\gamma = \frac{2}{\sigma^2}$ isolates the geometric components into their respective induced matrix norms, yielding

$$-\int_{\mathbb{R}^N}\left\langle \mathbf{A}(\theta;t)\nabla_\theta \ln\frac{\mu(t)}{\pi_t}, \mathbf{A}^{-1}(\theta;t)v_t\right\rangle \mu(t)\mathrm{d}\theta \le \frac{\sigma^2}{4}\mathcal{I}_{\mathbf{A}}(\mu(t)\parallel\pi_t)+\frac{1}{\sigma^2}\mathbb{E}_{\mu(t)}\left[\|v_t\|^2_{\mathbf{A}^{-2}(\theta;t)}\right]. \tag{45}$$

Combining these bounds gives the following inequality for the time derivative of the KL divergence.

$$\frac{d}{dt}\,\mathrm{KL}(\mu(t)\parallel\pi_t) \le -\frac{\sigma^2}{4}\mathcal{I}_{\mathbf{A}}(\mu(t)\parallel\pi_t)+\frac{1}{\sigma^2}\mathbb{E}_{\mu(t)}\left[\|v_t\|^2_{\mathbf{A}^{-2}(\theta;t)}\right]. \tag{46}$$

Integrating the differential inequality over the temporal interval $[0,T]$, dropping the nonnegative term $\mathrm{KL}(\mu(T)\parallel\pi_T)$, and rearranging to isolate the cumulative relative Fisher Information completes the proof.

□

B.3 Proof of Corollary 3.6

Proof of Corollary 3.6. Substituting the block-diagonal structure of the inverse preconditioning matrix $\mathbf{A}^{-2}(\theta;t)=\mathrm{diag}(\alpha_\phi^{-2}(\phi;t)\mathbf{I}_\phi,\alpha_W^{-2}(W;t)\mathbf{I}_W)$ yields

$$\mathbb{E}_{\mu(t)}\left[\|v_t\|^2_{\mathbf{A}^{-2}(\theta;t)}\right]=\mathbb{E}_{\mu(t)}\left[\frac{\|v_t^\phi(\theta)\|^2}{\alpha_\phi^2(\phi;t)}\right]+\mathbb{E}_{\mu(t)}\left[\frac{\|v_t^W(\theta)\|^2}{\alpha_W^2(W;t)}\right]. \tag{47}$$

Substituting the instantaneous transport costs $c_t^\phi(\theta)$ and $c_t^W(\theta)$ into the differential inequality from Theorem 3.5 and integrating over $[0,T]$ gives the blockwise bound after dropping the nonpositive terminal contribution $-\mathrm{KL}(\mu(T)\parallel\pi_T)$.

□

B.4 Conditional Allocation under a Fixed Gain Budget

Remark B.2 (Allocation under the generalized bound). Corollary 3.6 yields a per-block decomposition in which transport costs enter with inverse-quadratic gain scaling. To define a conditional allocation problem, fix positive block costs C_l, representing frozen values of $\int_0^T \mathbb{E}_{\mu(t)}[c_t^l]\,\mathrm{d}t$, independently of the candidate gains. For positive constant gains and a fixed budget $B>0$, with no additional gain constraints, consider

$$\min_{\alpha_l>0,\,\sum_l\alpha_l=B}\ \sum_l \frac{C_l}{\alpha_l^2}, \qquad \alpha_l^*=B\frac{C_l^{1/3}}{\sum_j C_j^{1/3}}, \quad l,j\in\{\phi,W\}. \tag{48}$$

The objective is strictly convex for positive costs, and its stationarity condition $2C_l/\alpha_l^3=\nu$, with budget multiplier ν, yields the stated unique minimizer. This is an optimum of the frozen drift-cost penalty, not necessarily of the complete bound: the initial KL divergence and the measures defining the transport costs may depend on the gains. It also minimizes the complete frozen surrogate if the initial KL term is held fixed. Moreover, the relative Fisher metric on the left-hand side changes with the gating matrix, so minimizing the surrogate does not establish optimal actual tracking or predictive performance.

If some costs vanish while others are positive, the same expression describes a boundary infimum approached as gains on zero-cost blocks tend to zero; it is not attained under strict positivity. If all costs vanish, every feasible allocation minimizes the penalty. Additional lower or upper gain constraints require a constrained allocation rather than the unrestricted formula above.

The conditional allocation motivates assigning larger scalar gains to higher-cost blocks. It does not directly identify an optimal Output/Hidden/Full mode: utility-gated layers need not receive larger gains than layers using fixed scaling. These mode comparisons are empirical tests motivated by the principle, not direct realizations of its surrogate optimum. The distributional transport cost is

not directly observable; checkpoint displacement from an ungated baseline is used only as a heuristic empirical proxy for relative block drift. Connecting that proxy to the transport-cost functional requires additional tracking and identifiability assumptions.

B.5 Scope of the Regularity Assumptions

The continuous-time results hold under the regularity assumptions stated in the main text. The stationary utility analysis assumes a stationary density, finite expectations in the utility identities, and vanishing boundary terms in integration by parts. The distributional tracking analysis additionally requires the stated measure and transport regularity, initial integrability, and finite transport cost.

When weight decay is positive and the loss is bounded below, the regularized potential is coercive, meaning that it grows without bound as the parameter norm increases. This property alone does not establish all the stationary-density and boundary conditions required for the gated diffusion. The MuJoCo experiments use zero weight decay, and bounded trajectories observed during finite training do not establish these global conditions.

The supervised networks use ReLU activations, and PPO uses a clipped surrogate, so the implemented objectives do not directly satisfy the global smoothness assumptions. The implemented gates also depend on running utility averages and maximum-based normalization. Although sigmoid gates are bounded, this does not guarantee smooth dependence on the current parameters or the uniform positive lower bound required for tracking. The theory therefore studies a smooth diffusion model under explicit regularity conditions rather than asserting that every experimental trajectory satisfies them.

C Extended Related Work

This section reviews explanations and measurements of plasticity loss, along with methods for maintaining plasticity and reducing forgetting. We then discuss layer-wise adaptation and its connections to utility gating, distributional tracking, and curvature.

C.1 Plasticity loss and its measurement

Prior training can affect both generalization and the ability to learn from new data (Ash & Adams, 2020; Berariu et al., 2021). Studies of plasticity loss have examined dormant neurons (Sokar et al., 2023), increasingly sparse activations and diminishing gradients (Abbas et al., 2023), reduced representation rank (Kumar et al., 2021; Lyle et al., 2022), and changes in loss curvature (Lyle et al., 2023; Lewandowski et al., 2023; He et al., 2025). Continued training can also produce declining learning performance in supervised and reinforcement-learning settings (Dohare et al., 2024). These observations motivate studying several mechanisms rather than attributing every instance of plasticity loss to a single network property.

Structural measurements require careful interpretation. Empirical work on offline reinforcement learning questions a simple relationship between feature rank and performance (Gulcehre et al., 2022). More recently, Wang et al. (2026) construct examples in which representation rank and neural tangent kernel rank fail to predict trainability. Their optimization readiness measure combines gradient strength and reliability and lower-bounds expected one-step optimization gain under smoothness, unbiased-gradient, and step-size conditions. This distinguishes an observed network property from a predictor with a stated performance guarantee.

Reinforcement learning introduces changing state distributions and bootstrapped targets even within a fixed environment (Klein et al., 2024). Primacy bias describes over-reliance on early experience (Nikishin et al., 2022), while studies of on-policy learning examine how plasticity loss and its remedies depend on the training setting (Juliani & Ash, 2024). Tang et al. (2025) connect plasticity loss to changes in predictions on samples outside the current training batch and propose reducing this churn. Their functional perspective complements measurements of network structure.

C.2 Methods for maintaining plasticity and reducing forgetting

Perturbation and selective reinitialization seek to sustain the ability to learn. Shrink and Perturb combines weight shrinkage with random perturbations to address the warm-start generalization gap (Ash & Adams, 2020). Continual Backprop replaces low-utility units (Dohare et al., 2021; 2024), while ReDo reinitializes dormant neurons (Sokar et al., 2023). Hernandez-Garcia et al. (2026) compare weight and unit reinitialization and identify settings, including small networks and networks with layer normalization, where weight reinitialization is more effective. These approaches differ in the granularity and criterion used to select which parts of the network are renewed.

Importance-based methods instead seek to preserve learned knowledge. Elastic Weight Consolidation uses Fisher-based importance estimates (Kirkpatrick et al., 2017), Synaptic Intelligence accumulates contributions to loss reduction (Zenke et al., 2017), and Memory Aware Synapses uses output sensitivity (Aljundi et al., 2018). Other regularizers address continued trainability. Regenerative regularization draws parameters toward their initial values (Kumar et al., 2023), distributional regularization constrains weight distributions (Lewandowski et al., 2023), and spectral regularization controls layer weight spectral norms (Lewandowski et al., 2025). These interventions motivate examining both what is modified and where the modification is applied.

C.3 Utility gating and layer-selective adaptation

Transfer-learning research shows that feature transferability varies across network depth (Yosinski et al., 2014). A survey relates memorization to generalization (Wei et al., 2024), while experiments with data augmentation show that useful earlier-layer representations can coexist with random-label memorization in later layers (Anagnostidis et al., 2023). These findings motivate studying layer-wise adaptation, but do not establish which layers should receive utility gating under a particular distribution change. Work on sparse subnetworks (Frankle & Carbin, 2019), low-rank adaptation (Hu et al., 2022), and adapter modules (Houlsby et al., 2019) provides related examples of restricting or structuring adaptation.

UPGD uses parameter utility to modulate gradient updates and perturbations, addressing both forgetting and plasticity loss (Elsayed & Mahmood, 2024). Its continuous update modulation differs from selectively reinitializing units or weights. Our work examines where this modulation is beneficial by comparing output-only, hidden-only, and full gating while continuing to update all layers. The theoretical analysis separates representation and head drift contributions, and measured block movement provides context for the empirical comparisons. Utility-score concentration does not by itself establish which parameters require protection, and the location of an input or target change does not uniquely determine the best gating placement.

C.4 Connections to tracking and curvature

Online optimization studies adaptation to changing objectives (Zinkevich, 2003; Besbes et al., 2015). Dynamic regret compares accumulated loss with that of a changing comparator. Our analysis instead bounds accumulated relative Fisher information between the evolving parameter distribution and a moving target distribution through the initial KL divergence and a transport cost. Relating this distributional guarantee to prediction-loss regret requires additional assumptions.

Loss-landscape sensitivity also motivates optimization methods such as Sharpness-Aware Minimization, which optimizes loss under local parameter perturbations (Foret et al., 2021). Our stationary utility theorem relates expected displacement-based utility to gated curvature and a correction for non-equilibrium probability currents. This quantity is distinct from UPGD's implemented gradient-weight utility. The empirical utility–curvature diagnostics therefore provide complementary evidence rather than a direct test of the theorem or a universal identification of high utility with high curvature.

D Synthetic Tracking Unit Test

We examine scalar-gain allocation in an additive two-block regression model with Gaussian observation noise:

$$y_t = (x_t^\phi)^\top \phi_t + (x_t^W)^\top W_t + \epsilon_t, \qquad \epsilon_t \sim \mathcal{N}(0, \sigma_\epsilon^2).$$

The ground-truth blocks evolve according to $\phi_{t+1} = \gamma_\phi \phi_t + \zeta_{\phi,t}$ and $W_{t+1} = \gamma_W W_t + \zeta_{W,t}$, with $\gamma_\phi = \gamma_W = 1$ (random-walk drift) and Gaussian innovations $\zeta_{\phi,t} \sim \mathcal{N}(0, \sigma_\phi^2 I)$ and $\zeta_{W,t} \sim \mathcal{N}(0, \sigma_W^2 I)$. These blocks are additive regression coefficients, not a composed representation and readout. The construction allows their innovations to be varied separately; it does not by itself guarantee identifiability or independent learning signals. Identifiability requires a nonsingular joint feature second-moment matrix. For squared loss, population cross-block coupling vanishes when $\mathbb{E}[x_t^\phi (x_t^W)^\top] = 0$, although sample gradients can still share a prediction residual.

Under a fixed total gain budget $\alpha_\phi + \alpha_W = \alpha$ ($\alpha = 0.02$, $d_\phi = d_W = 20$, $T = 500$, $\sigma_\phi, \sigma_W \in \{0, 0.01\}$, $\sigma_\epsilon = 0.1$, 30 seeds, learners initialized at the true parameters), we compare allocations of 25/75, 50/50, and 75/25 in representation/head order. The configurations vary whether innovations are applied to the head block, the representation block, or both. Zero innovation variance implies a constant block only when its decay and initialization also keep it fixed; equal innovation variances alone do not imply equal drift costs. The conditional allocation calculation in Section B.4 motivates assigning more gain to the block with greater fixed drift cost and equal gains when those costs are equal. We examine these qualitative predictions through test mean squared error.

At each step, the two feature blocks are sampled independently from standard normal distributions, with 2,048 training examples and 512 test examples. Learners share the data and ground-truth trajectory within each seed. Per-block gradient norms are clipped at 10, and parameter coordinates are clipped to $[-100, 100]$.

Figure 3 and Table 4 summarize test MSE averaged over the final 50 of 500 steps. Allocating more gain to the changing block gives lower error in the two single-block conditions. Equal gains give the lowest mean error when both blocks change in the evaluated symmetric configuration. These results support the qualitative allocation principle motivated by the conditional analysis and provide a controlled reference for the neural network gating comparisons that follow.

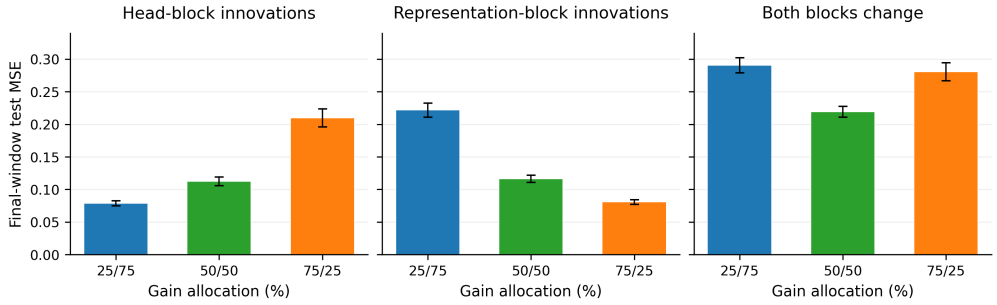


Figure 3: Synthetic tracking with a fixed total gain budget of 0.02. Columns show innovations in the head block, representation block, or both. Allocations give representation/head percentages of the gain budget. Bars show mean test MSE over the final 50 steps, with one sample-SD SE across 30 seeds, using the summaries in Table 4. No smoothing is applied.

Table 4: Synthetic tracking results over 30 seeds. Each seed contributes its mean test MSE over the final 50 steps. SE uses the sample standard deviation across seeds. Two-sided paired t-tests compare each allocation with the lowest-mean allocation within its condition. The reported p-values are unadjusted for multiple comparisons.

Innovations	Allocation	α_ϕ	α_W	Mean MSE	SE	p vs lowest mean
Head	25/75	0.005	0.015	0.0785	0.0039	—
Head	50/50	0.010	0.010	0.1126	0.0067	< 0.0001
Head	75/25	0.015	0.005	0.2097	0.0139	< 0.0001
Representation	75/25	0.015	0.005	0.0807	0.0037	—
Representation	50/50	0.010	0.010	0.1163	0.0057	< 0.0001
Representation	25/75	0.005	0.015	0.2217	0.0109	< 0.0001
Both	50/50	0.010	0.010	0.2191	0.0082	—
Both	75/25	0.015	0.005	0.2805	0.0137	< 0.0001
Both	25/75	0.005	0.015	0.2905	0.0115	< 0.0001

E Relative Block Movement and Utility–Curvature Diagnostics

E.1 Relative Block Movement from Training Checkpoints

We summarize parameter movement in baseline runs using saved checkpoints. For block $l \in \{\phi, W\}$, concatenate its parameters into θ_l and define $D_l = \sum_k \|\theta_l(t_{k+1}) - \theta_l(t_k)\|_2$. With block parameter count N_l, the normalized representation-to-head ratio is

$$\rho = \frac{D_\phi/\sqrt{N_\phi}}{D_W/\sqrt{N_W}}, \qquad D_W > 0. \tag{49}$$

The parameters are pooled within each block before taking a norm, and each block is normalized once. Smaller ratios indicate relatively less representation movement, and larger ratios indicate relatively more representation movement. The diagnostic describes observed optimizer trajectories rather than the moving target distribution's transport cost in the tracking bound. Architecture, parameterization, optimizer, and checkpoint spacing can affect its value.

Table 5 reports the supervised and MuJoCo measurements. The supervised probes use SGD, and the MuJoCo probes use Adam. In MuJoCo, the representation block contains the actor and critic hidden layers, and the head block contains both final linear layers. Actor log-standard-deviation parameters are excluded from both blocks. Each MuJoCo seed has 199 checkpoints at approximately 100K-step intervals, giving 198 consecutive displacements.

Table 5: Normalized representation-to-head checkpoint-displacement ratios across four supervised and four MuJoCo settings. Entries give the reported mean and 95% confidence-interval half-width across seeds, with 5–6 supervised seeds and five MuJoCo seeds. The measurements provide descriptive context for gating comparisons and do not define a threshold for selecting a mode.

Setting	Intervention	Baseline	ρ (95% CI)
CIFAR-10	Label permutation	SGD	0.055 ± 0.001
EMNIST	Label permutation	SGD	0.134 ± 0.002
Mini-ImageNet	Label permutation	SGD	0.145 ± 0.001
Input-Permuted MNIST	Input permutation	SGD	0.848 ± 0.017
Humanoid-v4	Standard single task	Adam	1.064 ± 0.027
HumanoidStandup-v4	Standard single task	Adam	1.304 ± 0.035
Walker2d-v4	Standard single task	Adam	1.559 ± 0.048
Ant-v4	Standard single task	Adam	1.583 ± 0.028

Within the supervised group, the label-permutation benchmarks have lower ratios than Input-Permuted MNIST. The MuJoCo measurements provide complementary context for the gating comparisons. These observations motivate examining representation and head movement together, without establishing a universal selection rule. In particular, the ratio alone does not explain the close Input-MNIST performance results.

For Gridworld, the pooled block ratios measured from Adam runs are 0.490 under reward shift, 0.530 under joint shift, and 0.570 under dynamics shift. They use ten pre-shift checkpoints from 5K to 50K episodes, yielding nine intervals per seed, with ten reward-shift seeds, eight joint-shift seeds, and ten dynamics-shift seeds. The hidden block contains 14,656 parameters and the output block contains 260. Their narrow ratio range supports relative comparisons within Gridworld, without supplying transferable thresholds across domains.

Exploratory comparison of shift composition. An exploratory EMNIST comparison gives ratios near 0.138 under label permutation and 0.151 when label and pixel permutations are mixed. Output-only gating exceeds full gating by approximately 0.038 accuracy in the label-only condition, but falls below it by approximately 0.098 in the mixed condition. The gating comparison uses three seeds per condition, while the ratios use five-seed SGD probes. Accuracy is substantially lower in the mixed condition, so these comparisons do not isolate shift composition from task difficulty. They motivate prospective evaluation across settings rather than a universal threshold for selecting a gating mode. The block tracking bound permits both blocks to drift, but does not establish the checkpoint ratio as a sufficient statistic for choosing where to gate.

E.2 Empirical Utility-Curvature Diagnostics

Theorem 3.2 relates the stationary expected displacement-based utility rate to gated curvature and a non-equilibrium probability-current correction. The implemented raw utility $u_i = -g_i\theta_i$ is a different quantity. We measure its Spearman correlations with the diagonal Hessian and squared gradient as empirical diagnostics. Figure 4 reports Gridworld comparisons, and Table 6 extends the measurements to supervised and MuJoCo settings. We denote Spearman correlation by r_s to distinguish it from the checkpoint ratio ρ.

UPGD-nonprotecting leaves gradients ungated but retains utility-dependent modulation of injected noise. Adam uses adaptive moment-based updates. Neither baseline represents the theorem's identity-gate diffusion. The correlations provide complementary information about the implemented utility signal without directly testing the stationary identity or identifying its probability-current correction.

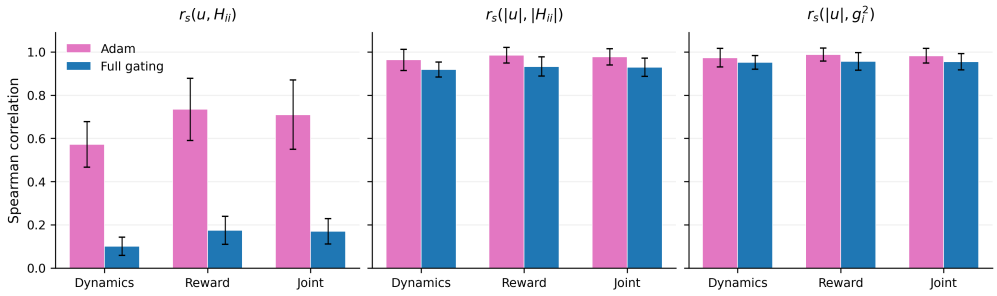


Figure 4: Spearman correlations in Gridworld DQN under Adam and full gating. Panels show $r_s(u, H_{ii})$, $r_s(|u|, |H_{ii}|)$, and $r_s(|u|, g_i^2)$. Bars and error bars give the mean and population SD over checkpoint observations pooled across seeds, not across-seed SE. Each condition includes 60 observations per seed. Adam uses 10 dynamics, 10 reward, and 8 joint-shift seeds, while full gating uses 10 per condition.

Table 6: Spearman correlations between raw utility magnitude and Hessian magnitude or squared gradient. Supervised and MuJoCo entries show the mean and population SD of within-seed checkpoint averages across five seeds. Gridworld entries give the range of condition means across the three interventions. These descriptive statistics are not tests of the theoretical diffusion's identity-gate case.

Setting	Method	$r_s(\|u\|, \|H\|)$	$r_s(\|u\|, g^2)$	Seeds
Gridworld DQN	Adam	0.96–0.98	0.97–0.99	8–10
Gridworld DQN	Full gating	0.92–0.93	0.95–0.96	10
EMNIST	UPGD-nonprotecting	0.934 ± 0.006	0.981 ± 0.002	5
EMNIST	Full gating	0.505 ± 0.024	0.826 ± 0.009	5
Mini-ImageNet	UPGD-nonprotecting	0.924 ± 0.003	0.976 ± 0.001	5
Mini-ImageNet	Full gating	0.452 ± 0.020	0.822 ± 0.007	5
Input-MNIST	UPGD-nonprotecting	0.830 ± 0.002	0.948 ± 0.001	5
Input-MNIST	Full gating	0.652 ± 0.013	0.888 ± 0.005	5
CIFAR-10	UPGD-nonprotecting	0.293 ± 0.013	0.786 ± 0.005	5
CIFAR-10	Full gating	0.223 ± 0.009	0.757 ± 0.003	5
Ant-v4	Adam	0.37 ± 0.05	0.84 ± 0.02	5
Ant-v4	Full gating	0.62 ± 0.02	0.87 ± 0.01	5
Walker2d-v4	Adam	0.39 ± 0.06	0.88 ± 0.01	5
Walker2d-v4	Full gating	0.58 ± 0.05	0.89 ± 0.01	5
Humanoid-v4	Adam	0.61 ± 0.01	0.91 ± 0.004	5
Humanoid-v4	Full gating	0.63 ± 0.01	0.90 ± 0.01	5
HumanoidStandup-v4	Adam	0.73 ± 0.02	0.94 ± 0.005	5
HumanoidStandup-v4	Full gating	0.82 ± 0.03	0.95 ± 0.01	5

The reported utility-magnitude correlations with squared gradient are high across the measured settings, while the curvature correlations vary more substantially across datasets and optimizers. The signed and magnitude correlations in Gridworld also differ. These observations support studying the information carried by implemented utility without identifying it universally with curvature or with displacement-based utility.

F Supervised Experimental Setup and Utility Analysis

We describe the supervised continual-learning setup and present additional analyses of utility distributions, layer-wise utility concentration, and the effect of utility clamping.

F.1 Experimental Setup

We follow the continual supervised-learning benchmarks of Elsayed & Mahmood (2024). Label assignments change every 2,500 updates on Mini-ImageNet, EMNIST, and CIFAR-10. Input-Permuted MNIST changes the input pixel permutation every 5,000 updates while preserving class labels. Each run contains 10^6 updates with batch size one. Mini-ImageNet supplies precomputed ResNet-50 features, while the other benchmarks supply flattened image inputs. The trained network has two hidden ReLU layers of widths 300 and 150 and a linear output layer. All layers update under every gating configuration. Final accuracy is computed by averaging task accuracy over the last 40 tasks within each run before aggregating across runs.

F.1.1 Run Records for Final Accuracy

The following counts describe cached run records used in the final-accuracy panels of Figure 1. Each record contributes its mean accuracy over the last 40 tasks. The cache does not retain seed identities, so these counts do not establish numbers of independent seeds or pairing across methods.

Table 7: Run counts for the final-accuracy panels of Figure 1. Counts refer to cached run records used for each dataset and method.

Dataset	Output-only gating	Full gating	Hidden-only gating	S&P
Mini-ImageNet	26	26	8	5
EMNIST	8	8	7	5
CIFAR-10	12	12	11	5
Input-MNIST	12	12	11	5

F.2 Per-Layer Utility Statistics

Table 8 summarizes the reported high-score tail mass and maximum utility magnitude for the output layer and the hidden-layer average.

Table 8: Reported per-layer utility statistics across the supervised benchmarks. Tail mass is the percentage of parameters with transformed utility score $s > 0.52$. Raw utility maxima are reported separately from the transformed-score tail mass.

Dataset	Layer	Tail Mass (> 0.52)	Raw$u_{\max}$	Output/Hidden Ratio
CIFAR-10	linear_3	1.06–1.08%	0.42–0.51	3.8–5.0×
CIFAR-10	hidden avg	0.22–0.28%	0.21–0.30	–
EMNIST	linear_3	0.24–0.39%	0.38–0.52	2.3–5.4×
EMNIST	hidden avg	0.07–0.11%	0.09–0.15	–
Mini-ImageNet	linear_3	0.31–0.42%	0.29–0.48	1.3–2.1×
Mini-ImageNet	hidden avg	0.20–0.24%	0.07–0.13	–
Input-MNIST	linear_3	17.7–18.6%	0.47–0.52	76–98×
Input-MNIST	hidden avg	0.19–0.24%	0.15–0.18	–

The output layer has a larger proportion of high-score parameters and higher reported maximum utility magnitudes than the hidden-layer average across all four benchmarks. This concentration motivates examining utility modulation at the output layer. The accuracy comparisons then evaluate whether extending that modulation to hidden layers provides additional benefit.

F.3 Utility Distributions and Gate Selectivity

Figure 5 compares utility-score distributions across the three gating variants. Differences in their central mass and tails show how gating placement affects the distribution of utility scores during training. We examine these distributions alongside accuracy to better understand the behavior of each variant.

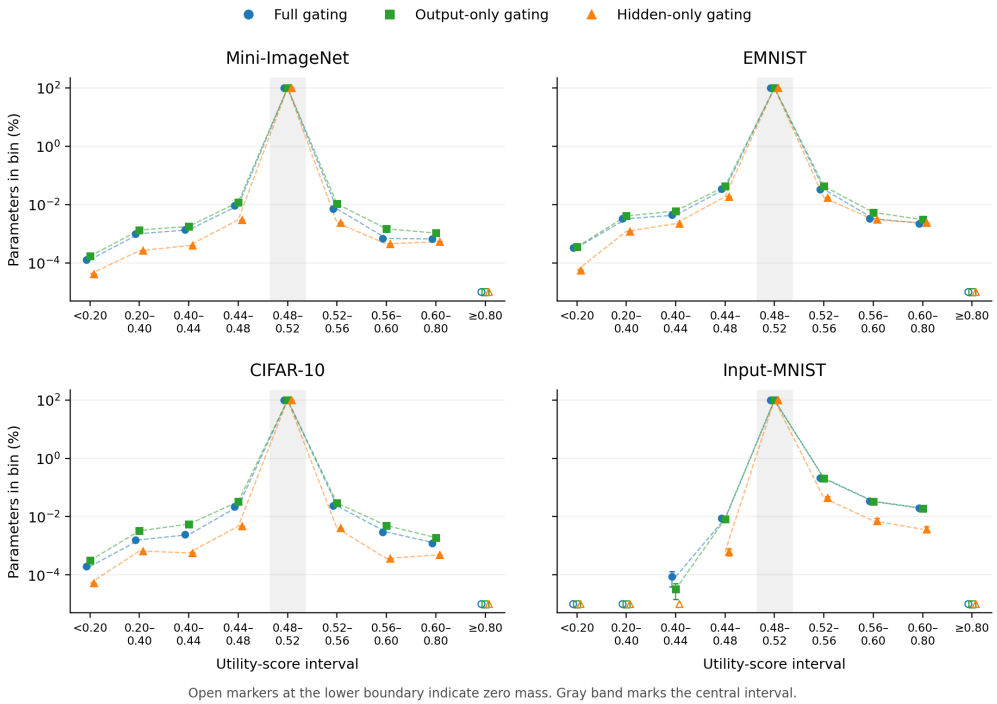


Figure 5: Distribution of implemented utility scores across four supervised continual-learning benchmarks and three gating configurations. Points show mean percentages of parameters in each score bin, with error bars indicating one standard error across runs. The vertical axis is logarithmic.

For parameters receiving utility gating, the multiplier satisfies $a = 1 - s$. Scores above 0.52 therefore give multipliers below 0.48, while scores below 0.48 give multipliers above 0.52. Relative to the neutral value $a = 0.5$, these multipliers respectively reduce or increase both the gradient and injected-noise components. Weight decay is applied separately. Global normalization and the sigmoid can place many scores near $s = 0.5$, making the score tails useful for describing how utility modulation varies across parameters. The clamping experiment tests sensitivity to changing this distribution.

F.4 Accuracy under Utility Clamping

Table 9 and Figure 6 compare accuracy under four clamping conditions. Every reported clamp reduces accuracy relative to the unclamped condition, with the largest reductions under the narrowest interval. The intermediate conditions are not uniformly ordered across datasets.

Table 9: Accuracy under utility clamping, expressed as percentages. The upper-cap condition limits utility scores to at most 0.52.

Clamp condition	Mini-ImageNet (%)	EMNIST (%)	CIFAR-10 (%)
Unclamped	61.50	72.71	59.55
Upper cap at 0.52	45.55	70.46	55.40
[0.44, 0.56]	40.10	54.26	56.05
[0.48, 0.52]	30.29	37.11	52.48

Clamping changes the gate distribution, so these results show sensitivity to utility modulation without isolating the causal contribution of the extreme tails or attributing differences between datasets to task complexity.

G Gating Coverage and Computational Cost

Table 10 reports the parameter counts and the fraction receiving utility-dependent modulation in each layer-selective variant.

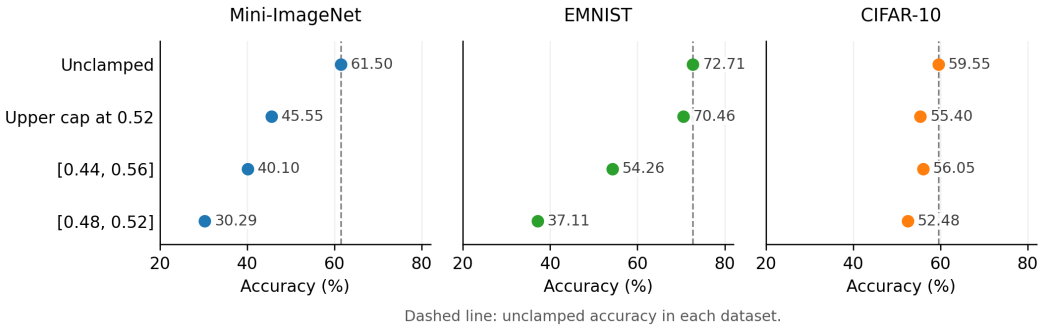


Figure 6: Accuracy under utility clamping on Mini-ImageNet, EMNIST, and CIFAR-10. Points show the values in Table 9. Dashed vertical lines mark each dataset's unclamped accuracy. All panels share the same accuracy scale, and clamp conditions are displayed as separate categories. No uncertainty intervals are shown.

Table 10: Parameter counts and gated fraction for each layer-selective variant.

Dataset	Total Params	Output Layer	Output-only gating (%)	Hidden-only gating (%)
Mini-ImageNet	674,950	15,100	2.2%	97.8%
EMNIST	287,747	7,097	2.5%	97.5%
CIFAR-10	968,560	1,510	0.16%	99.8%
Input-MNIST	282,160	1,510	0.54%	99.5%

Output-only gating applies utility-dependent modulation to a small fraction of the network's parameters. However, the evaluated implementation tracks utility across all parameters because their utilities contribute to the global normalizer. Utility tracking therefore requires $\mathcal{O}(N)$ computation per update and $\mathcal{O}(N)$ storage in every gating mode, where N is the total number of updated parameters. The gated fractions describe where modulation acts and do not establish corresponding runtime or memory savings.

H Diagnostic Metrics for Dead Units and Plasticity

We report dead-unit fractions and the short-horizon plasticity proxy $1 - \ell_{\text{after}}/\ell_{\text{before}}$ as complementary diagnostics. Tables 11 and 12 give the reported summaries across datasets and methods. Figures 7 and 8 show S&P, output-only gating, and full gating.

Table 11: Reported dead-unit fractions for S&P and the two UPGD-W gating variants across the supervised benchmarks.

Method	Mini-ImageNet	EMNIST	CIFAR-10	Input-MNIST
S&P	0.9464	0.9431	0.7876	0.7942
Output-only gating	0.8633	0.8141	0.7876	0.5231
Full gating	0.7668	0.7318	0.7799	0.5891

Table 12: Reported short-horizon plasticity for S&P and the two UPGD-W gating variants across the supervised benchmarks.

Method	Mini-ImageNet	EMNIST	CIFAR-10	Input-MNIST
S&P	0.4119	0.4849	0.3799	0.5773
Output-only gating	0.5069	0.4327	0.3876	0.4961
Full gating	0.5558	0.5065	0.4125	0.5177

These diagnostics do not reproduce the final-accuracy ordering across methods. On the evaluated label-permutation benchmarks, output-only gating has higher reported final accuracy than full gating even though the aggregate diagnostics do not consistently favor it. This difference motivates reporting the diagnostics alongside performance. It does not establish that either metric is uninformative in other settings or that gating location causally outweighs every other measure of plasticity.

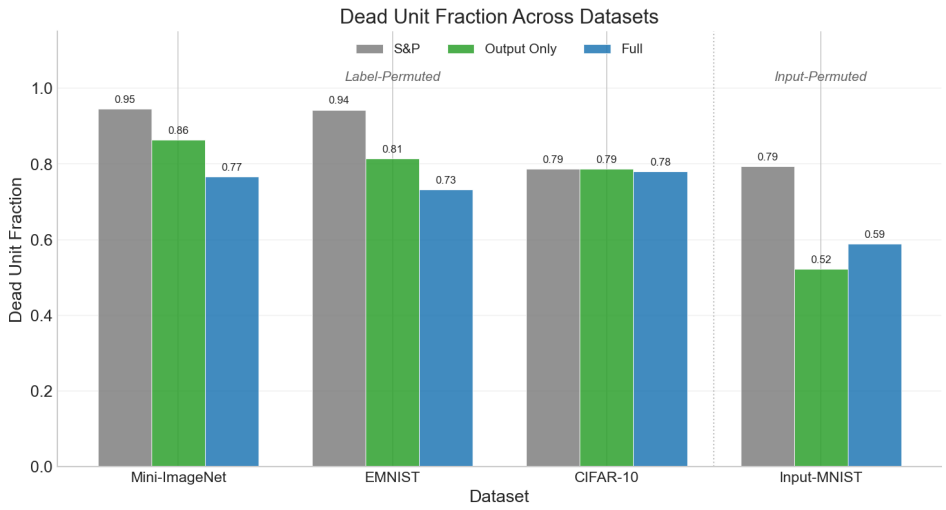


Figure 7: Dead-unit fractions for S&P, output-only gating, and full gating on four supervised continual-learning benchmarks. The first three benchmarks use label permutation, while Input-MNIST uses input permutation. Bar labels show the reported fractions rounded to two decimal places.

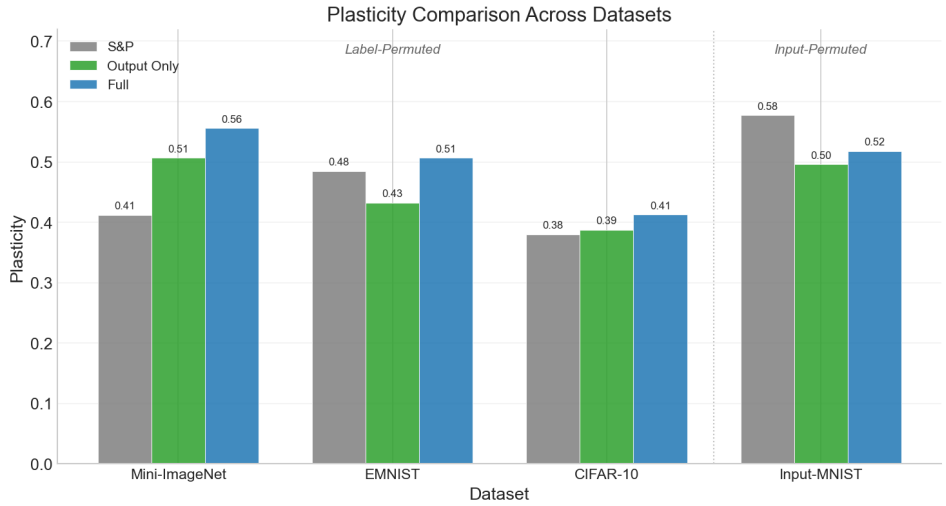


Figure 8: Short-horizon plasticity for S&P, output-only gating, and full gating. The proxy $1 - \ell_{\text{after}}/\ell_{\text{before}}$ measures relative loss reduction over an adaptation interval. Bar labels show the reported values rounded to two decimal places.

I Reinforcement Learning Evaluation Details

I.1 Gridworld

We use a 7×7 Gridworld to compare utility-gating placements under controlled changes to the goal location, obstacle layout, or both. Goal changes affect termination, visitation, and action-value targets. Obstacle changes affect transitions and Bellman targets while retaining the one-hot position encoding. These interventions let us examine how gating placement affects adaptation to different environment changes. Their effects on learned representations and action-value targets can overlap.

Environment. Observations are 49-dimensional one-hot encodings of position. The agent selects from four directional actions, receives -0.1 per transition and $+1.0$ upon reaching the goal, and terminates an episode at the goal or after 200 transitions.

Shift conditions. The environment changes every 5,000 episodes, giving 10 phases over 50,000 episodes (Table 13).

Table 13: Shift conditions used in Gridworld. Environmental changes occur every 5,000 episodes.

Condition	Environmental change
Reward shift	Goal location
Dynamics shift	Obstacle layout
Joint shift	Goal location and obstacle layout

Shift configurations. We use five goal positions and five obstacle layouts. Figure 9 illustrates the first change under each shift condition, and Tables 14 and 15 specify all configurations. No goal position overlaps an obstacle in any of the 25 goal-layout combinations. Coordinates are zero-based (row, column), with row 0 at the top and column 0 at the left.

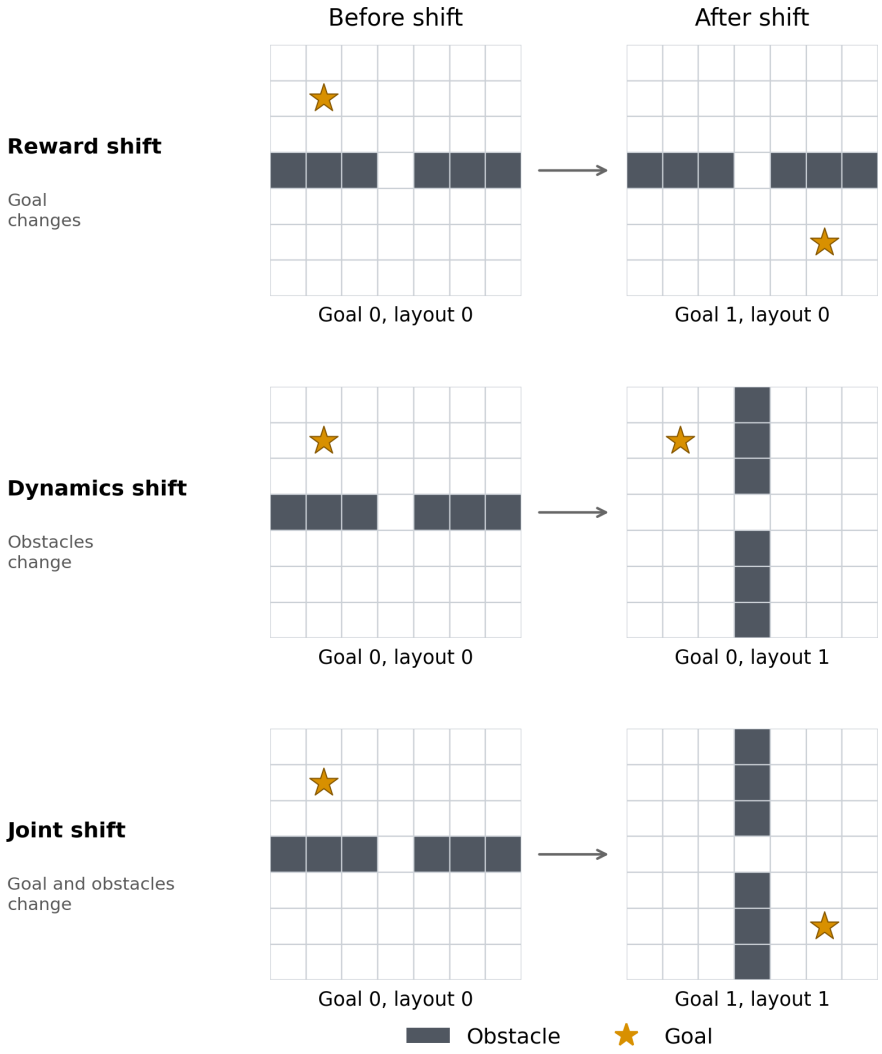


Figure 9: Examples of the first configuration change in the 7×7 Gridworld. Columns show the environment before and after the shift from phase 0 to phase 1. Reward shift changes the goal while retaining the obstacle layout, dynamics shift changes the obstacles while retaining the goal, and joint shift changes both. Dark cells indicate obstacles and stars indicate goals. Configuration indices correspond to Tables 14 and 15. Agent starting positions and trajectories are omitted.

Goal positions.

Obstacle layouts.

Phases are indexed from $k = 0$, and each phase lasts 5,000 episodes. Under reward shift, the goal uses index $k \bmod 5$ while the obstacle layout remains at index 0. Under dynamics shift, the layout uses index $k \bmod 5$ while the goal remains at index 0, or $(1, 1)$. Under joint shift, both use

Table 14: Goal positions used in Gridworld. Coordinates are zero-based (row, column).

Index	Spatial Position	Description
0	$(1, 1)$	Near top-left
1	$(5, 5)$	Near bottom-right
2	$(1, 5)$	Top-right area
3	$(5, 1)$	Bottom-left area
4	$(4, 5)$	Center-right

Table 15: Obstacle layouts used in Gridworld. Listed coordinates identify blocked cells using zero-based (row, column) indexing.

Index	Pattern	Blocked Cell Coordinates
0	Horizontal wall with gap	$(3, 0), (3, 1), (3, 2), (3, 4), (3, 5), (3, 6)$
1	Vertical wall with gap	$(0, 3), (1, 3), (2, 3), (4, 3), (5, 3), (6, 3)$
2	Short horizontal barrier and isolated obstacle	$(2, 4), (2, 5), (2, 6), (4, 2)$
3	Scattered obstacles	$(0, 2), (2, 0), (4, 4), (6, 2), (2, 6)$
4	Diagonal barrier with gap	$(0, 5), (1, 4), (2, 3), (4, 1), (5, 0)$

index $k \bmod 5$. The active configuration sequence therefore repeats every five phases. All methods encounter the same intervention sequence, although visitation and episode lengths can differ.

Network and training. The Q-network is a ReLU MLP with dimensions $49 \to 128 \to 64 \to 4$ and 14,916 parameters. The output layer has 260 parameters, or 1.7% of the network. Online training uses batch size 1 without replay, learning rate 10^{-3}, and discount factor $\gamma = 0.99$. An ε-greedy policy decreases ε from 1.0 to 0.05 over the first 1,000 episodes and resets it to 0.3 after each shift. The target network is synchronized every 100 episodes. Training horizons and synchronization are matched in episodes. Differences in episode length can produce different numbers of transitions and optimizer updates across methods.

UPGD-W settings. We use utility smoothing $\beta_{\text{utility}} = 0.999$, noise scale $\sigma = 0.001$, weight decay 0.001, and fixed nongated scale 0.5. Adam is compared with full gating, output-only gating of L3, and hidden-only gating of L1 and L2. Each condition uses 10 seeds per method, giving 120 online runs.

Evaluation metrics. We measure plateau return over the final 500 episodes of each phase and early-window return over the first 500 episodes after each shift. Each window contains 50 recorded individual episode returns. We average the window means across post-shift phases within each run, excluding the initial phase, and report the mean and standard error across 10 runs using the sample standard deviation.

Online results over 50K episodes. Figure 2 and Table 2 in the main text show the online learning curves and plateau returns. Table 16 provides the detailed plateau returns. Under dynamics shift, hidden-only gating has a higher mean plateau return than output-only gating, with full gating between them. Output-only gating has the highest mean under reward shift, while full gating has the highest mean under joint shift. These observations support including hidden layers in utility gating when adapting to dynamics changes.

Table 16: Online Gridworld plateau returns over 50K episodes, reported as mean $\pm$ sample-based SE across 10 runs. Each run averages the final 500-episode windows of nine post-shift phases.

Condition	Method	Plateau $\pm$ SE
Reward shift	Adam	-5.258 ± 1.286
Reward shift	Hidden-only gating	0.116 ± 0.315
Reward shift	Output-only gating	0.511 ± 0.006
Reward shift	Full gating	0.376 ± 0.100
Dynamics shift	Adam	0.556 ± 0.004
Dynamics shift	Hidden-only gating	0.555 ± 0.004
Dynamics shift	Output-only gating	0.507 ± 0.012
Dynamics shift	Full gating	0.530 ± 0.010
Joint shift	Adam	-3.567 ± 0.830
Joint shift	Hidden-only gating	0.544 ± 0.004
Joint shift	Output-only gating	0.536 ± 0.009
Joint shift	Full gating	0.550 ± 0.004

Table 17 reports early-window returns separately. Their ordering should not be inferred from plateau performance.

Table 17: Online Gridworld early-window returns over 50K episodes. Values are mean ± sample-based SE across 10 runs. Each run averages the first 500-episode windows across nine post-shift phases. This endpoint measures initial adaptation performance rather than recovery time.

Condition	Method	Early-window return ± SE
Reward shift	Output-only gating	-17.665 ± 0.086
Reward shift	Full gating	-17.398 ± 0.258
Reward shift	Hidden-only gating	-16.992 ± 0.262
Reward shift	Adam	-16.391 ± 0.270
Dynamics shift	Output-only gating	-1.594 ± 0.154
Dynamics shift	Full gating	-1.449 ± 0.173
Dynamics shift	Hidden-only gating	-1.076 ± 0.082
Dynamics shift	Adam	-1.075 ± 0.097
Joint shift	Output-only gating	-15.958 ± 0.212
Joint shift	Full gating	-16.271 ± 0.209
Joint shift	Hidden-only gating	-15.802 ± 0.110
Joint shift	Adam	-14.964 ± 0.293

Extended reward-shift training and layer gating. We extend reward-shift training to 100,000 episodes, or 20 phases. The four main methods are evaluated together with L1-only, L2-only, and L2+L3 gating, giving 70 runs across seven methods and 10 seeds. Figure 10 shows all seven methods, and Table 18 reports plateau returns averaged over 19 post-shift phases.

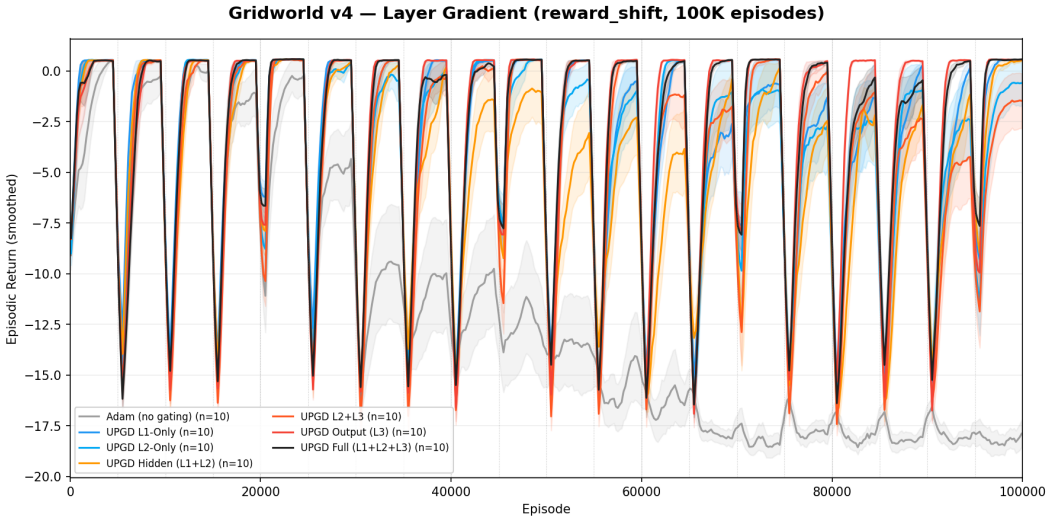


Figure 10: Gridworld reward-shift training over 100K episodes for seven methods. Recorded individual returns are averaged within left-closed 200-episode bins in each run. Across-seed means and sample-based SEs are each smoothed with a five-bin moving average with edge replication, without resetting at phase boundaries. Curves show the smoothed mean and shading shows the correspondingly smoothed SE across 10 seeds. Dotted lines mark changes every 5,000 episodes. These epsilon-greedy training returns are not independent policy evaluations.

The additional variants gate L1 ($49 \to 128$), L2 ($128 \to 64$), or L2 and L3 together, where L3 maps $64 \to 4$. The experiment does not show a monotonic improvement as gating moves toward the output layer. Output-only gating has the highest mean plateau return, followed by full gating.

L1-only has a positive mean of 0.077 ± 0.217, while L2+L3 gating has a negative mean. The results therefore do not establish a general advantage for every configuration that includes the output layer or identify a causal explanation based on the hidden layers' representational role. Figure 11 shows the endpoint comparison.

Replay buffer ablation. We repeat the 50K-episode evaluation with a replay buffer of 10,000 transitions and batch size 64. Replay reuses earlier experience, making this a different training setting from the online experiment (Nikishin et al., 2022). Figure 12 shows the learning curves.

Table 18: Gridworld reward-shift plateau returns over 100K episodes. Values are mean ± sample-based SE across 10 runs, after averaging the final 500-episode windows of 19 post-shift phases within each run.

Condition	Method	Plateau ± SE
Reward shift	Adam	-11.535 ± 0.796
Reward shift	L1	0.077 ± 0.217
Reward shift	L2	-0.439 ± 0.414
Reward shift	Hidden-only gating	-0.688 ± 0.448
Reward shift	L2+L3	-0.328 ± 0.363
Reward shift	Output-only gating	0.514 ± 0.005
Reward shift	Full gating	0.371 ± 0.067

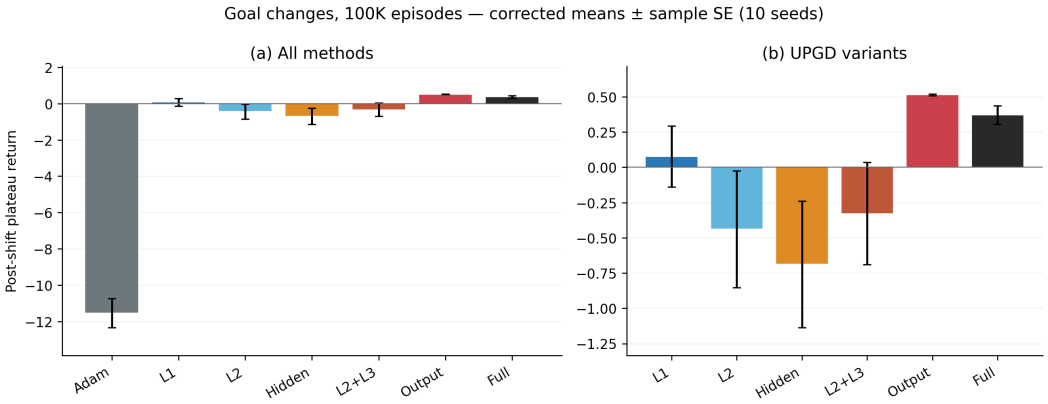


Figure 11: Gridworld reward-shift plateau returns over 100K episodes. Within each run, returns are averaged over the final 500-episode windows of 19 post-shift phases. Bars show means across 10 runs, with error bars indicating one standard error. The left panel includes all methods, and the right panel shows the UPGD-W variants.

Mean plateau returns are close across methods within each shift condition. Table 19 reports plateau returns, while Table 20 reports early-window returns.

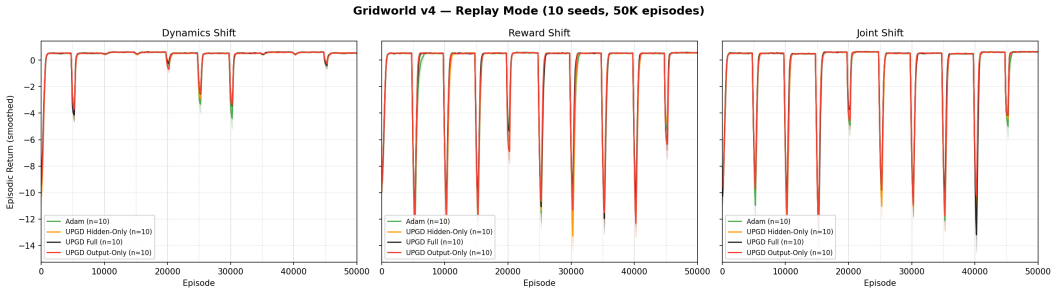


Figure 12: Gridworld replay training over 50K episodes under dynamics shift, reward shift, and joint shift. Recorded individual returns are averaged within left-closed 100-episode bins in each run. Across-seed means and sample-based SEs are separately smoothed with a five-bin moving average with edge replication, without resetting at phase boundaries. Curves show the smoothed mean and shading shows the correspondingly smoothed SE across 10 seeds. Dotted lines mark changes every 5,000 episodes. These epsilon-greedy training returns are not independent policy evaluations.

I.2 MuJoCo

We evaluate PPO on Ant-v4, Walker2d-v4, Humanoid-v4, and HumanoidStandup-v4. The fixed-gating comparisons run for 20 million environment steps and compare Adam with output-only, hidden-only, and full utility gating. These are standard single-task benchmarks without imposed shifts. Table 21 reports returns averaged over the final 2 million steps.

Table 19: Gridworld replay plateau returns over 50K episodes. Values are mean ± sample-based SE across 10 runs, after averaging the final 500-episode windows of nine post-shift phases within each run.

Condition	Method	Plateau ± SE
Reward shift	Adam	0.530 ± 0.004
Reward shift	Hidden-only gating	0.533 ± 0.004
Reward shift	Output-only gating	0.529 ± 0.004
Reward shift	Full gating	0.532 ± 0.004
Dynamics shift	Adam	0.564 ± 0.003
Dynamics shift	Hidden-only gating	0.554 ± 0.006
Dynamics shift	Output-only gating	0.557 ± 0.005
Dynamics shift	Full gating	0.562 ± 0.003
Joint shift	Adam	0.554 ± 0.004
Joint shift	Hidden-only gating	0.556 ± 0.003
Joint shift	Output-only gating	0.558 ± 0.004
Joint shift	Full gating	0.552 ± 0.004

Table 20: Replay Gridworld early-window returns over 50K episodes. Values are mean ± sample-based SE across 10 runs. Each run averages the first 500-episode windows across nine post-shift phases. This endpoint measures initial adaptation performance rather than recovery time.

Condition	Method	Early-window return ± SE
Reward shift	Output-only gating	-10.613 ± 0.274
Reward shift	Full gating	-10.109 ± 0.205
Reward shift	Hidden-only gating	-10.760 ± 0.210
Reward shift	Adam	-10.690 ± 0.235
Dynamics shift	Output-only gating	-0.977 ± 0.055
Dynamics shift	Full gating	-0.995 ± 0.076
Dynamics shift	Hidden-only gating	-0.978 ± 0.129
Dynamics shift	Adam	-1.213 ± 0.083
Joint shift	Output-only gating	-9.537 ± 0.194
Joint shift	Full gating	-9.605 ± 0.261
Joint shift	Hidden-only gating	-9.975 ± 0.189
Joint shift	Adam	-10.247 ± 0.215

PPO optimizer. The PPO evaluations use a moment-normalized UPGD variant that maintains first and second gradient moments, analogous to Adam. Separately, the utility proxy $\tilde{u}_{t,i} = -g_{t,i}\theta_{t,i}$ is tracked with a bias-corrected exponential moving average. This signal is normalized by the signed global maximum of the uncorrected moving-average utility and passed through a logistic sigmoid to obtain $s_{t,i} \in (0, 1)$. On gated parameters, $1 - s_{t,i}$ scales both the normalized gradient update and injected noise. Nongated parameters use a fixed scale of 0.5. All actor and critic parameters continue to update under every gating configuration.

Network and training. All four tasks use separate actor and critic MLPs with two 64-unit hidden layers and $\tanh$ activations. PPO uses learning rate 3×10^{-4}, discount $\gamma = 0.99$, GAE parameter $\lambda = 0.95$, rollout length 2048, 10 update epochs, clipping coefficient 0.2, and gradient norm clipping at 0.5. UPGD-W uses $\beta_{\text{utility}} = 0.999$, noise scale $\sigma = 0.001$, zero weight decay, and nongated scale 0.5.

Gating configurations that include hidden layers have the highest mean return in all four tasks. On Ant-v4, hidden-only gating has a slightly higher mean return than full gating, with an unresolved difference in the approximate comparison below. Full gating has the highest mean on Walker2d-v4, Humanoid-v4, and HumanoidStandup-v4. These results support retaining full gating as a practical candidate alongside output-only gating in continuous control. All layers continue to update and receive perturbations in every variant. Neither observation dimension nor action dimension alone identifies a task's drift structure.

Table 22 relates these comparisons to measured relative block movement. The ratios provide context rather than a fixed rule for choosing a gating placement.

Ant-v4 evaluation. Ant-v4 uses six seeds for Adam and full gating, 38 for output-only gating, and 42 for hidden-only gating.

Table 21: MuJoCo PPO returns averaged over the final 2 million of 20 million environment steps, reported as mean ± SE across runs. Numerical ordering does not establish significant separation among all gating variants.

Environment	Method	Runs	Mean return ± SE
Ant-v4	Adam	6	748 ± 236
Ant-v4	Full gating	6	3892 ± 705
Ant-v4	Hidden-only gating	42	4544 ± 169
Ant-v4	Output-only gating	38	2989 ± 148
Walker2d-v4	Adam	20	2893 ± 124
Walker2d-v4	Full gating	20	3685 ± 86
Walker2d-v4	Hidden-only gating	20	3618 ± 81
Walker2d-v4	Output-only gating	20	3468 ± 118
Humanoid-v4	Adam	20	576 ± 12
Humanoid-v4	Full gating	20	1077 ± 89
Humanoid-v4	Hidden-only gating	20	733 ± 13
Humanoid-v4	Output-only gating	20	717 ± 14
HumanoidStandup-v4	Adam	20	100343 ± 4069
HumanoidStandup-v4	Full gating	20	137696 ± 3971
HumanoidStandup-v4	Hidden-only gating	20	124897 ± 3735
HumanoidStandup-v4	Output-only gating	20	126361 ± 4765

Table 22: Gating comparisons across four single-task MuJoCo benchmarks and measured representation-to-head displacement ratios. The HumanoidStandup-v4 comparison uses an approximate, unadjusted Welch test. Measurement details and uncertainty for ρ are reported in Appendix E.

Environment	Measured ρ	Gating with highest mean return	Comparison notes
Ant-v4	1.583	Hidden-only gating	Hidden-only gating has a slightly higher mean than full gating. Their approximate pairwise comparison is not significant.
Walker2d-v4	1.559	Full gating	Pairwise differences among gating variants are not statistically resolved.
Humanoid-v4	1.064	Full gating	Full gating achieves a higher mean than both selective variants.
HumanoidStandup-v4	1.304	Full gating	Full versus output-only gating gives approximate $p = 0.076$.

Hidden-only and full gating have means of 4544 ± 169 and 3892 ± 705, respectively. An approximate two-sided Welch test reconstructed from these displayed means, SEs, and seed counts gives $p = 0.406$. This unadjusted comparison does not resolve their difference.

Phase-adaptive gating on Ant-v4. We extend Ant-v4 training to 40M steps to compare fixed gating with schedules that switch placement at equal thirds of training. The evaluation includes nine UPGD-W schedules and Adam, with five seeds each. We use h, f, and o for hidden-only, full, and output-only gating. For example, $h \to h \to f$ applies hidden-only gating for the first two thirds and full gating for the final third. These schedules empirically test whether changing placement improves performance. The tracking bound does not identify a theoretically optimal schedule for Ant-v4.

Table 23 reports return over the final 4M steps. These runs form a separate five-seed cohort from the 20M evaluation.

Uniform hidden-only gating has the highest mean return of 5273 ± 325. Its differences from $h \to h \to o$ at 5020 ± 280 and uniform full gating at 4576 ± 138 are not significant in approximate unadjusted two-sided Welch tests, with $p = 0.57$ and $p = 0.10$, respectively. No pair among the eight hidden-first or full-first schedules reaches the nominal 0.05 level in tests reconstructed from the displayed means and SEs with five seeds per schedule. In particular, $h \to f \to h$ is a hidden-first schedule with a lower mean than every full-first schedule. Uniform hidden-only gating differs significantly from uniform output-only gating and Adam in these approximate unadjusted comparisons. These comparisons do not establish a benefit from switching gating placement during training.

Walker2d-v4 setup. Walker2d-v4 has a 17-dimensional observation and six-dimensional action space. We use the shared PPO architecture and hyperparameters described above, with 20 seeds per method over 20M steps.

Table 23: Phase-adaptive gating on Ant-v4 over 40M steps. Values are mean return over the final 4M steps ± SE across five runs per schedule. Pairwise tests discussed in the text are approximate and unadjusted.

Schedule	Description	Mean Return± SE
h→h→h	Uniform hidden-only gating	5273 ± 325
h→h→o	Hidden, then output late	5020± 280
h→f→f	Hidden early, full mid+late	4936± 270
h→h→f	Hidden early+mid, full late	4904± 275
f→h→h	Full early, hidden mid+late	4802± 141
f→f→f	Uniform full gating	4576± 138
f→f→h	Full early+mid, hidden late	4497± 235
h→f→h	Hidden-full-hidden sequence	4480± 370
o→o→o	Uniform output-only gating	3265± 333
Adam	Adam baseline	44± 55

Walker2d-v4 results. Full, hidden-only, and output-only gating all outperform Adam in the reported comparisons, with $p \leq 0.002$. Their pairwise differences are not statistically resolved, with reported $p = 0.58$ for full versus hidden-only gating, $p = 0.30$ for output-only versus hidden-only gating, and $p = 0.15$ for output-only versus full gating. The numerical means are 3685, 3618, and 3468, respectively (Table 21). Observation dimension alone does not explain the contrast with Ant-v4.

Humanoid-v4 setup. Humanoid-v4 has a 376-dimensional observation and 17-dimensional action space. We use the shared PPO architecture and hyperparameters, with 20 seeds per method over 20M steps.

Humanoid-v4 results. Full gating achieves the highest mean final-window return of 1077 ± 89, approximately 87% above Adam's 576 ± 12. Hidden-only and output-only gating achieve 733 ± 13 and 717 ± 14, respectively (Table 21). Their reported pairwise comparison is not significant at $p = 0.42$. The result supports including full gating among the placements considered for continuous control. It does not identify a joint-shift mechanism from the observation and action dimensions, and an Adam return plateau alone does not diagnose plasticity loss without a separate test of continued trainability.

HumanoidStandup-v4 setup. HumanoidStandup-v4 uses the same 376-dimensional observation and 17-dimensional action space as Humanoid-v4, with the task of standing from a prone position. We use the shared PPO architecture and hyperparameters, with 20 seeds per method over 20M steps.

HumanoidStandup-v4 results. Full gating has the highest mean return of 137696 ± 3971, compared with 126361 ± 4765 for output-only gating and 124897 ± 3735 for hidden-only gating (Table 21). All three improve on Adam in the reported baseline comparisons. Approximate unadjusted two-sided Welch tests reconstructed from the displayed means and SEs give $p = 0.076$ for full versus output-only gating and $p = 0.024$ for full versus hidden-only gating. Only the latter reaches the nominal 0.05 level, without multiplicity adjustment.

J Scope and Limitations

The supervised experiments use two-hidden-layer ReLU networks trained online with batch size one under abrupt label or input permutations. Mini-ImageNet uses precomputed ResNet-50 features, so its feature extractor is fixed. The findings concern these architectures and the evaluated hyperparameter configurations. Utility clamping changes the broader gate distribution and therefore does not isolate the contribution of extreme utility tails.

Gridworld evaluates adaptation to imposed environmental changes, while MuJoCo uses standard single-task benchmarks. Gridworld training is matched in episodes, but differences in episode length can lead to different numbers of transitions and optimizer updates across methods. Its comparisons do not isolate independent representation and head drift. In MuJoCo, the same gating choice is applied to the actor and critic, and Ant-v4 uses unequal run counts across methods. These features limit how broadly the observed comparisons can be generalized.

The continuous-time analysis holds under the stated regularity, stationary-utility, and measure-transport assumptions. The tracking bound controls accumulated relative Fisher information.

Checkpoint displacement describes observed parameter movement, and its connection to the target-distribution transport cost remains unestablished. The conditional allocation calculation holds with fixed block costs and a fixed total gain budget. Gating placement is therefore evaluated empirically, and the measured ratio is used as descriptive context rather than a universal selection rule.

Future work can extend these comparisons to deeper end-to-end networks, gradual distribution shifts, and separate gating choices for the actor and critic.