NEURON SHARING IN MULTI-TASK REINFORCEMENT LEARNING POLICIES FOLLOWS THE GEOMETRY OF THE TASK PARAMETERIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-task RL policies are usually treated as monolithic, yet prior work shows subnetworks for individual behaviors exist within them. We investigate how these subnetworks are structured within the network to identify neuron reusage and specificity. Using gradient-based importance profiles and causal ablation on policies trained with Proximal Policy Optimization, we measure neuron sharing between tasks. In a continuously parameterized locomotion task (AntDir), neuron sharing decays monotonically with angular distance between task goal directions, consistently across seeds and network widths. Removing one task's top neurons preferentially degrades its angular neighbors. This organization persists when all continuous goal features are replaced by one-hot task IDs, so it is learned from the behavioral geometry of the task family rather than inherited from the input encoding. In HalfCheetahVel, overlap also tracks velocity distance, but ablation damage is nonselective, showing that geometry-aligned overlap does not imply functional modularity. On a fully learned seven-task Meta-World subset, which lacks a continuous task parameterization, the ordering of pairwise overlap barely replicates across seeds and ablation effects are asymmetric. Together, these results indicate that neuron sharing in these policies follows the geometry of the task parameterization when one exists, and that overlap alone does not establish functional modularity.

1 INTRODUCTION

Multi-task reinforcement learning (MTRL) trains a single policy to solve multiple tasks simultaneously. A central finding of this field is that multi-task training produces shared internal representations that generalize across tasks and facilitate transfer to new ones (Parisotto et al., 2016; Johnston & Fusi, 2021; Vafidis et al., 2025). These representations have been characterized at the level of representational geometry as disentangled, abstract, or factored structures (Johnston & Fusi, 2021; Feather & Chung, 2023; Vafidis et al., 2025). However, a more granular question remains open: at the level of individual neurons, which units are reused across which tasks, and what determines the pattern of sharing? Understanding this internal organization could inform architecture design, network pruning, and the composition of sub-policies for transfer.

Mechanistic interpretability provides tools for investigating such questions. Recent work has identified circuits and subnetworks responsible for specific behaviors in supervised models (Marks et al., 2025; Hadad et al., 2026; Melander et al., 2026; Chowdhury et al., 2026) and, increasingly, in RL agents (Mini et al., 2023; Trim & Grayston, 2024). However, these studies focus on isolating circuits for individual behaviors within a single model. They do not address how these task-specific subnetworks overlap in a consistent way across seeds and different environments.

To answer the question of overlapping structure, we develop a three-stage analysis pipeline. First, we define task-specific neuron importance profiles using gradient$\times$activation attribution on the penultimate layer Melander et al. (2026), capturing each neuron's causal contribution to the policy output under each task condition. Second, we measure pairwise neuron sharing between tasks via the Bhattacharyya coefficient on L1-normalized importance profiles. Third, we causally validate the importance profiles by ablating each task's most important neurons and measuring collateral dam-

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.

age to all other tasks. We apply this pipeline to three environments: AntDir, HalfCheetahVel, and Meta-World, spanning continuously parameterized and discrete task families. To isolate the role of task geometry, we hold the learning setup fixed, using PPO-trained MLP policies with two hidden layers throughout, and deliberately vary task geometry, goal encoding, and network width.

Our main finding is that neuron sharing in the network reflects the geometry of the task family. Neuron importance overlap decays monotonically with angular distance between task directions in AntDir and with velocity distance in HalfCheetahVel. This angular organization persists when all continuous goal features are replaced by a one-hot task ID, so it is not inherited from a geometric input encoding. Instead, the continuous goal code acts as an inductive bias that sharpens the gradient, most visibly when network capacity is scarce. Ablation preferentially damages neighboring tasks in AntDir, whereas HalfCheetahVel exhibits geometry-aligned overlap without corresponding ablation selectivity. On a fully learned seven-task Meta-World subset, where tasks lack a continuous parameterization, the ordering of pairwise overlap is only weakly reproducible across seeds and ablation damage is asymmetric. Together, these results suggest that in the policies we study the behavioral geometry of the task family shapes the sharing patterns learned within the multi-task policy network.

2 Related Work

MTRL methods have progressed from training single policies via expert distillation (Parisotto et al., 2016) to scalable distributed architectures (Espeholt et al., 2018) and reward normalization techniques (Hessel et al., 2018) that enable a single agent to master diverse task sets. A consistent finding is that multi-task training produces abstract or disentangled representations that support generalization (Johnston & Fusi, 2021; Vafidis et al., 2025). Some architectures impose modularity explicitly through policy sketches (Andreas et al., 2017), shared value spaces (Borsa et al., 2016), or routing networks (Rosenbaum et al., 2017). We instead ask whether modularity emerges in a standard Multilayer Perceptron. McLean et al. (2025b) recently showed that naively scaling a simple MTRL baseline outperforms sophisticated architectures, suggesting that task diversity matters more than architectural inductive biases. Our finding that internal sharing structure depends on task geometry rather than architecture is complementary. These works establish that shared representations exist, but do not characterize their neuron-level organization across tasks.

Mechanistic Interpretability identifies minimal circuits implementing specific behaviors. Automated circuit discovery systematizes activation patching to isolate causally relevant components (Conmy et al., 2023), with refinements including attribution patching (Syed et al., 2023). However, activation magnitude alone ignores how strongly a neuron drives the output. Melander et al. (2026) combine gradient and activation to capture the neuron's influence on the output. Still most of mechanistic interpretability work focuses on supervised models Melander et al. (2026); Marks et al. (2025); Huang & Hahn (2026); Chowdhury et al. (2026). The ones that focus on Reinforcement Learning look for circuit extraction rather than structural understanding Liu et al. (2026).

Work on understanding RL policy internals has identified goal-tracking circuits in maze-solving agents (Mini et al., 2023), revealed navigational biases via saliency mapping (Trim & Grayston, 2024), and shown that actor and critic representations specialize when separated (Garcin et al., 2025). The closest precedent to our work is the finding by Driscoll et al. (2024) that RNNs trained on multiple cognitive tasks reuse shared dynamical motifs across tasks, with reuse reflecting task similarity. We study the analogous question in feedforward RL policies, using neuron-level importance attribution rather than dynamical systems analysis, and focus on how the structure of the task parameterization determines the geometry of internal sharing.

3 Methods

3.1 Task-Conditioned Neuron Importance Profiles

We define the importance of a neuron to a given task via a gradient-based attribution that captures how much the neuron's activation influences the policy's output. Prior work has relied on activations alone to identify neuron-level influence on network behavior Kim et al. (2018). However, activations do not account for the downstream weight connections; a highly active neuron connected by near-zero weights contributes little to the output. We therefore use a gradient$\times$activation measure that

reflects both the neuron's activity and its causal effect on the action, which can also be called a neuron's *contribution* Melander et al. (2026).

We analyze neurons on the penultimate layer, the last hidden representation before the linear output head. At this layer the network has formed its highest-level features while remaining directly connected to the output, making neuron to action attribution straightforward. The same procedure applies to any layer without modification. Appendix F repeats the analysis on the first hidden layer. To compute importance, we collect rollouts across all tasks to a total of $100{,}000$ timesteps, divided equally among tasks ($\approx$12,500 per task). Episode lengths in our environments range from $\sim$500 (Meta-World) to $\sim$1000 (HalfCheetahVel), so this budget provides 13–25 full episodes per task. For each observation, we compute the importance of each neuron to the action produced by the policy. We then average these importance values across all observations from a given task condition to obtain a task-specific importance profile.

Let π_θ be a trained policy with penultimate-layer representation $z \in \mathbb{R}^H$. The policy outputs the mean of a Gaussian action distribution, $a = f(z) \in \mathbb{R}^D$; we compute importance with respect to this mean. For a single observation x_t, the Jacobian of the mean action with respect to the representation is:

$$J_t = \left.\frac{\partial a}{\partial z}\right|_{z_t} \in \mathbb{R}^{D\times H} \tag{1}$$

The importance of neuron h for action dimension d at timestep t is:

$$I_t[d,h] = \left.\frac{\partial a_d}{\partial z_h}\right|_{z_t} \cdot z_t[h] \tag{2}$$

This gradient$\times$activation product captures both how sensitive the output is to neuron h (the Jacobian) and how strongly neuron h is activated (the representation value). Because the output head is linear, at the penultimate layer J_t equals the head's weight matrix, so importance is the neuron's activation weighted by the strength of its readout. At earlier layers the Jacobian also depends on the state through downstream ReLU gating.

To obtain a single importance score per neuron, we take the mean absolute contribution across action dimensions:

$$\bar{I}_t[h] = \frac{1}{D}\sum_{d=1}^{D} |I_t[d,h]| \tag{3}$$

The absolute value ensures that a neuron contributing positively to one action dimension and negatively to another is recognized as influential in both, rather than having its contributions cancel. The resulting score $\bar{I}_t[h] \geq 0$ measures the total influence of neuron h on the action at timestep t. Appendix G shows that at the penultimate layer the absolute value discards no task-dependent sign information.

The importance profile for task k is the mean over all N_k timesteps collected under task condition k:

$$\mu_k[h] = \frac{1}{N_k}\sum_{t\in\mathcal{T}_k} \bar{I}_t[h] \tag{4}$$

where $\mathcal{T}_k$ is the set of timesteps from rollouts executed under task k. Since each $\bar{I}_t[h] \geq 0$, the profile $\mu_k \in \mathbb{R}^H_{\geq 0}$ is non-negative, with larger values indicating greater reliance on neuron h.

3.2 Importance Overlap Metrics

Once the profiles for each task are computed, we can use them to understand how the network organizes neuron usage across tasks. To measure that, we compute a pairwise similarity between the importance profiles of each task pair. The main metric we use is the Bhattacharyya Coefficient (BC), which quantifies the overlap between two distributions. Since the importance profiles are non-negative, we L1-normalize each profile to produce a probability distribution over neurons. For task k, the *importance distribution* is:

$$p_k[h] = \frac{\mu_k[h]}{\sum_{h'} \mu_k[h']}. \tag{5}$$

The Bhattacharyya coefficient between tasks i and j is then:

$$\mathrm{BC}(i,j) = \sum_{h=1}^{H} \sqrt{p_i[h] \cdot p_j[h]} \tag{6}$$

$\mathrm{BC} = 1$ when two tasks have identical importance distributions; lower values indicate more distinct neuron usage.

Each entry $p_k[h]$ is the fraction of total importance mass allocated to neuron h, and BC measures the overlap between two such allocations. We choose BC over cosine similarity because this distributional interpretation makes the overlap directly readable as a sharing fraction.

3.3 Causal Ablation Protocol

To validate that the importance profiles identify functionally relevant neurons, we perform causal ablation. For each task k, we rank neurons by their importance $\mu_k[h]$ in descending order and construct a binary mask that zeros out the top-s fraction of neurons in the penultimate layer during the forward pass. We call these *forward ablation masks*. We evaluate at sparsity levels $s \in \{0.05, 0.10, 0.20, 0.30, 0.50\}$, corresponding to ablating 5% to 50% of neurons. The main text reports $s = 0.05$; Appendix D shows that higher sparsities preserve the same qualitative patterns. For each mask, we evaluate the policy on all tasks using deterministic actions (the Gaussian mean) over 100 episodes per task and report mean episodic return. Comparing ablated returns against unmasked baselines reveals how damage to one task's important neurons affects that task and others.

If masking a task's most important neurons preferentially degrades that task and its close neighbors in task space, it provides causal evidence that the importance profiles identify functionally relevant neurons shared according to task geometry. Conversely, if ablating one task's neurons severely degrades that task but spares distant tasks, it indicates modular internal organization.

3.4 Environments and Training

The main environments we use to investigate geometry-respecting neuron sharing are AntDir and HalfCheetahVel, which feature continuous task parameterizations (direction and velocity, respectively). We also use Meta-World MT10 as a negative control, where tasks share an interface but lack a continuous parameterization.

The AntDir environment derives from the Ant environment in MuJoCo Schulman et al. (2016); Towers et al. (2025), where instead of moving forward the policy must steer the ant toward a specific goal direction. We use 8 directions at 45 degree intervals. The goal direction is encoded as a 2D unit vector appended to the 105-dimensional Ant observation along with derived features (current velocity, velocity error, and heading error), yielding a 113-dimensional observation. The reward is the dot product of the ant's velocity with the goal direction, plus standard contact, control, and survival terms. Similarly, HalfCheetahVel derives from the HalfCheetah environment Wawrzyński (2009); Towers et al. (2025), where the task is to run at a specific target velocity. We use 8 velocities from 0.25 to 1.75 in increments of 0.25. The target velocity is appended as a scalar to the 17-dimensional base observation, yielding 18 dimensions. The reward is the negative absolute difference between current and target velocity, plus the standard control cost.

The Meta-World MT10 benchmark consists of 10 distinct manipulation tasks with different objects and goals McLean et al. (2025a). All tasks share a common robotic arm and a 39-dimensional proprioceptive observation space. The task identity is provided as a 10-dimensional one-hot vector appended to the observation (49 dimensions total). Because our MT10 policies do not learn all ten tasks, we also train on the subset of seven tasks they do learn (reach, door-open, drawer-open, drawer-close, button-press-topdown, window-open, window-close), with a 7-dimensional one-hot task identity (46 dimensions total) and otherwise identical settings.

On all environments the policy is an actor-critic architecture with two hidden layers of equal width and ReLU activations, with separate networks for actor and critic. We train with Proximal Policy Optimization (PPO) (Schulman et al., 2017) for 50 million steps on AntDir and HalfCheetahVel, and 20 million steps on MT10. The MT10 budget matches the 2 million steps per task used by McLean et al. (2025a). We attempted a longer training, but as Appendix I shows, doubling timesteps did not improve returns.

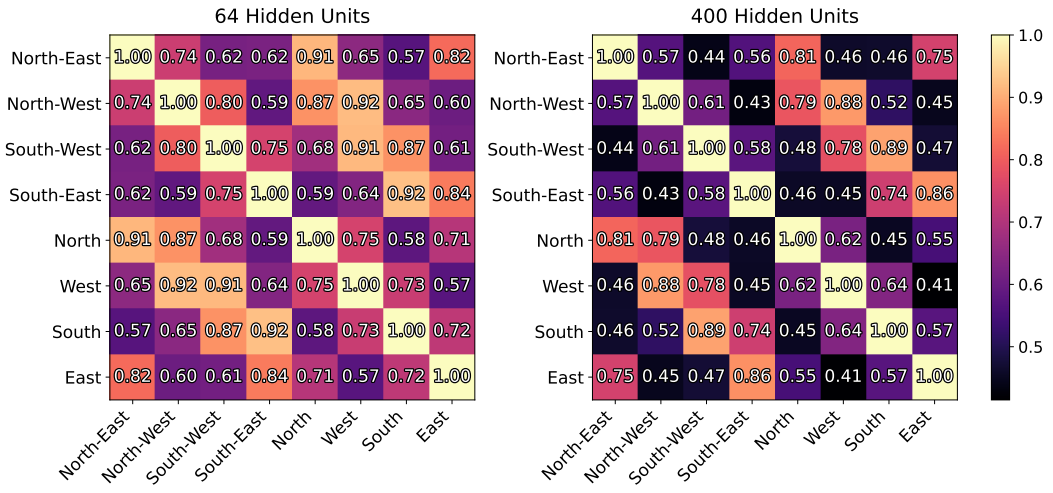

Figure 1: Mean Bhattacharyya Coefficient (BC) between neuron importance profiles for different task directions across all five seeds in the AntDir environment. On the left, the matrix for the experiment with 64 hidden units per layer; on the right, the matrix for 400 hidden units. Both show a clear monotonic decay of BC with angular distance, indicating geometry-respecting neuron sharing.

We run 5 random seeds per environment and two hidden-layer widths (64 and 400 units per layer) with the exception of HalfCheetahVel which we only run the 64 units variation. The wider architecture follows the network size for Meta-World MT10 in the current benchmarks McLean et al. (2025a). Full PPO hyperparameters are provided in Appendix A.

4 Experiments and Results

4.1 AntDir

Figure 1 shows a clear geometry-respecting structure in AntDir. BC decays monotonically with angular distance between tasks, indicating that the network preferentially reuses neurons for similar directions. For example, North has a higher BC with its 45° neighbors (NE, NW) than with its opposite (South), and this pattern holds across all directions and both widths. Table 1 summarizes mean BC at each angular distance. The decay also holds within every individual seed: mean BC decreases strictly from 45° to 180° in all five seeds at both widths.

Table 1: Mean BC between task pairs grouped by angular distance, for the continuous goal encoding and one-hot control, with random-initialization baselines. Values are mean ± std across the task pairs at each distance, computed on the seed-averaged BC matrix.

Angular distance	45°	90°	135°	180°
64 units, continuous	0.882±0.038	0.729±0.047	0.623±0.033	0.590±0.018
64 units, one-hot	0.897±0.076	0.798±0.063	0.733±0.045	0.702±0.010
400 units, continuous	0.812±0.054	0.588±0.029	0.469±0.020	0.435±0.014
400 units, one-hot	0.790±0.042	0.623±0.027	0.524±0.019	0.496±0.010
Random, continuous (64)	0.982±0.005	0.949±0.013	0.919±0.021	0.908±0.023
Random, continuous (400)	0.976±0.004	0.932±0.009	0.895±0.011	0.881±0.011
Random, one-hot (64)	0.970±0.009	0.969±0.012	0.968±0.010	0.968±0.013
Random, one-hot (400)	0.960±0.005	0.960±0.005	0.960±0.006	0.960±0.005

Table 1 also includes random-baseline rows, computed from networks with the same architecture and seeds but randomly initialized weights and no training. A mild angular gradient is visible even

there under the continuous encoding. This is expected, the goal direction is encoded as a 2D unit vector, so similar directions produce more correlated inputs, and any network, trained or not, will produce correlated activations in response. What distinguishes the trained networks is how sharply training amplifies this prior. The 64-unit network exhibits a total decay of 0.292, and the 400-unit network 0.376, roughly four times steeper than the input-driven baselines of 0.075 and 0.095 at the same widths. Training therefore amplifies, rather than merely forwards, the geometric structure present in the input.

To test whether the organization is learned or inherited from the input structure, we retrained the AntDir policies with all continuous goal features replaced by an 8-dimensional one-hot vector. We kept direction pool, hyperparameters, seeds, and training budget identical. Appendix B shows that the one-hot policies reach 98% and 99% of the continuous grand-mean return at widths 64 and 400, so the comparison is not confounded by training quality. Under one-hot input all task pairs are equidistant in input space, and the random-initialization baseline is accordingly flat in angular distance. Its decay is ≤ 0.003 at both widths as seen in Table 1. The trained one-hot networks nonetheless recover the monotonic angular decay. They present 0.195 at width 64 and 0.295 at width 400, positive in 5/5 seeds at both widths. The organization therefore cannot be inherited from the input encoding. At width 64 the decay is attenuated relative to the continuous encoding (0.195 vs. 0.292), suggesting the continuous goal code acts as an inductive bias that sharpens the organization when capacity is scarce, but is not its source.

Width also matters to neuron overlap. In Table 1, the 400-unit networks show lower sharing and a steeper angular decay than the 64-unit networks under both encodings. This format is consistent with wider networks specializing their profiles while keeping the shared structure, and complementing the finding that wider networks learn more tasks (McLean et al., 2025b).

The causal ablation results in Table 2 confirm that the shared neurons are functionally relevant, and that collateral damage tracks angular distance. Self-damage establishes that the targeted neurons are genuinely important. In the 64-unit continuous policies, each task retains on average only 38% of baseline return when its own top-5% neurons are removed. Collateral damage follows an angular gradient. Averaging across all source tasks, 45° neighbors retain 53% of baseline, 90° neighbors retain 67%, and 135° and 180° counterparts retain 73% and 72%. This 19 percentage-point spread between nearest and farthest neighbors mirrors the BC ordering.

Table 2: Retained return as % of baseline by angular distance between the ablated and evaluated task when ablating each task's top-5% neurons (mean ± std across the (ablated, evaluated) task pairs at each distance, computed on seed-averaged returns; 0° = self).

	Self (0°)	45°	90°	135°	180°
64 units, continuous	38.1±7.7	53.1±12.4	67.1±14.7	72.6±8.1	71.8±5.1
64 units, one-hot	34.0±10.8	40.2±12.6	49.5±11.8	54.5±15.5	56.7±17.4
400 units, continuous	51.4±3.2	69.7±13.0	84.7±9.4	90.3±8.8	94.7±6.3
400 units, one-hot	50.9±5.6	71.0±9.3	84.6±8.5	95.3±6.3	94.4±5.9

The subnetworks are not fully modular. At width 64, even the most distant tasks lose over a quarter of their return when another task's top neurons are ablated, so neurons that are most important for one task still matter for others. At width 400, distant tasks are nearly spared, consistent with the greater specialization of wider networks noted above. Together, the BC analysis and ablation establish that importance profiles identify genuinely shared neurons, and that the degree of sharing is organized by angular distance in task space.

Table 2 also shows that the functional organization survives the one-hot control. Under one-hot conditioning, ablating a task's top-5% neurons still damages that task the most, with collateral damage decreasing monotonically in angular distance. At width 400 the one-hot profile is indistinguishable from the continuous one. Self-damage matches (50.9 ± 5.6 vs. 51.4 ± 3.2) and the collateral gradient agrees cell-for-cell within noise. At width 64 the one-hot network keeps a comparable angular spread but retains less return at all distances, indicating more total sharing rather than a flattened gradient. The equal self-damage at matched sparsity also shows that the importance profiles are equally well calibrated under both encodings, so the profiles themselves are not an artifact of reading out the goal-vector inputs.

4.2 HALFCHEETAHVEL

In HalfCheetahVel, Figure 2 shows that importance overlap also decays with velocity distance, so the geometry-respecting structure is not unique to AntDir. However, Table 3 shows that ablating one velocity's important neurons degrades all velocities uniformly, with no preferential damage to the ablated velocity or its neighbors.

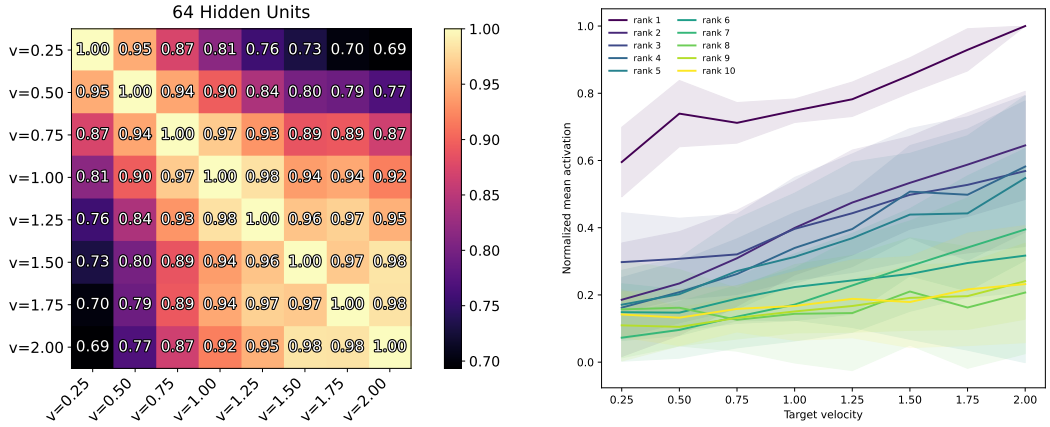

Figure 2: Left: mean Bhattacharyya Coefficient (BC) between neuron importance profiles for different target velocities in HalfCheetahVel. Right: mean activation of the ten most important penultimate-layer neurons versus target velocity. Since neuron indices differ between seeds, curves are aligned by importance rank and normalized by each seed's peak activation. Lines and bands show mean ± 1 std across five seeds.

Table 3: Difference between ablated return and unablated baseline. Rows correspond to the source velocity of the mask (ablated), and columns correspond to the target velocity (evaluated).

Ablated / Eval	0.25	0.50	0.75	1.00	1.25	1.50	1.75	2.00
0.25	-162±155	-197±183	-194±176	-207±209	-210±214	-315±415	-907±1952	-1625±3562
0.50	-140±92	-224±182	-223±147	-337±265	-449±315	-1032±1048	-1522±1958	-2082±2511
0.75	-137±84	-207±128	-258±132	-299±157	-387±200	-548±345	-1200±1404	-2399±3405
1.00	-132±80	-211±144	-279±151	-351±193	-460±307	-707±515	-1423±1372	-2672±3274
1.25	-133±79	-216±143	-303±142	-380±158	-512±274	-789±460	-1525±1248	-2828±3138
1.50	-134±79	-230±148	-312±144	-416±197	-508±273	-798±498	-1167±916	-1705±1488
1.75	-166±147	-232±151	-326±150	-405±175	-525±270	-865±561	-1038±707	-1508±1102
2.00	-173±145	-231±151	-321±151	-395±180	-512±274	-854±568	-1036±709	-1509±1101

Table 3 reports return difference rather than percentage of baseline because ablation frequently drives returns negative, making ratio-based metrics undefined. The column-wise gradient, where higher target velocities suffer larger absolute losses, is explained by the reward function. An agent whose behavior is degraded tends to slow down or stop, and the penalty for undershooting is proportional to the target velocity.

The critical observation is the uniformity across rows. For a given evaluation velocity, the return loss is nearly identical regardless of which velocity's neurons are ablated. This contrasts sharply with AntDir, where Table 2 shows self-damage exceeding collateral damage at every angular distance in all four configurations. In HalfCheetahVel, the diagonal carries no special penalty.

This dissociation between importance overlap and ablation selectivity reveals how the two measures capture different properties. BC quantifies how similarly two tasks weight the same neurons. It tracks task geometry in both environments. Ablation selectivity, however, requires that tasks rely on *distinct* neuron subsets, so that removing one task's neurons spares the others. In AntDir, steering in different directions demands qualitatively different leg coordination patterns. The network then allocates partially distinct neuron pools to different directions, and ablating one pool preferentially

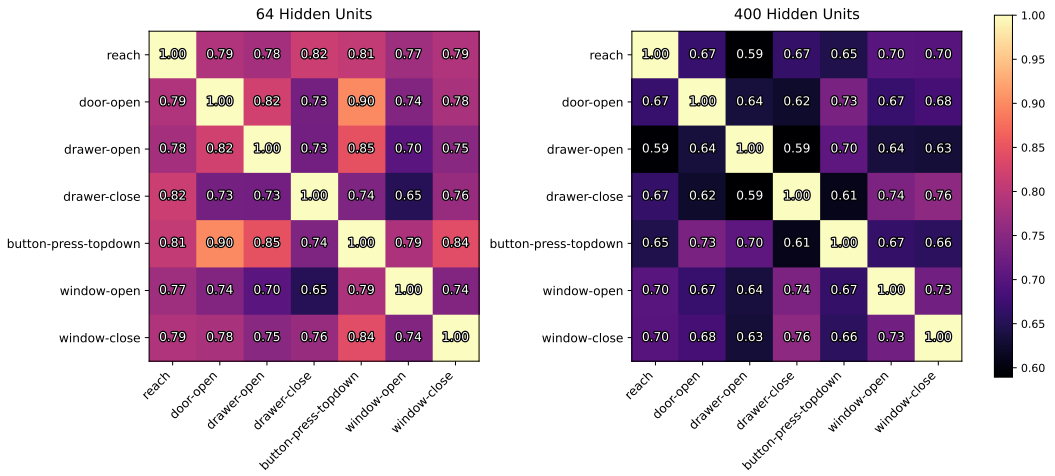

Figure 3: Mean Bhattacharyya Coefficient (BC) between neuron importance profiles for the seven-task Meta-World subset across all five seeds. On the left, the matrix for the experiment with 64 hidden units per layer; on the right, the matrix for 400 hidden units. Individual pair values do not replicate across seeds.

damages nearby directions that share it. In HalfCheetahVel, geometry-aligned overlap is not accompanied by preferential damage to the ablated velocity or its neighbors. This pattern is consistent with all velocities using the same running gait modulated by amplitude. The right panel of Figure 2 supports this. The ten most important neurons are active at every target velocity, and their activation increases with the target rather than switching between velocity-specific sets. The rank-1 neuron rises from 0.60 at $v = 0.25$ to 1.00 at $v = 2.00$, and lower ranks follow the same trend with more variability across seeds. Importance profiles still vary with velocity because the *magnitude* of each neuron's contribution changes, producing geometry-respecting BC. However, these ablations do not establish that all velocities use an identical neuron pool or rule out the existence of velocity-specific neurons.

4.3 META-WORLD

Our MT10 policies surpass random baselines on only seven of the ten tasks. So that any absent structure cannot be attributed to absent learning, we retrained on these seven tasks with identical architecture, hyperparameters, and training budget. The retrained policies learn all seven tasks at both widths, and Appendices C and H report per-task returns for both Meta-World task sets.

Figure 3 shows the BC matrices and Table 4 summarizes them. Sharing is moderate and roughly uniform across task pairs. The mean off-diagonal BC is close to the random-initialization level at width 64 and below it at width 400. The few pairs with slightly higher overlap visually differ between widths. Therefore we tested whether the pair ordering was stable across seeds by computing the Spearman correlation of the 21 pairwise BC values between every pair of seeds. The ordering is only weakly reproducible, with a mean ρ of 0.21 at width 64 and 0.30 at width 400. The same statistic on AntDir is 0.59 and 0.82 with the continuous encoding and 0.38 and 0.76 with the one-hot encoding. In AntDir, this agreement comes entirely from angular distance where seeds consistently rank closer directions as more similar, but order pairs at the same angle inconsistently. Meta-World has no such task distance to anchor the ordering, so individual high- or low-overlap pairs there should not be read as task relations.

The ablation results in Table 5 contrast sharply with AntDir in two ways. First, self-damage is not privileged. For three of the seven tasks, reach, drawer-open, and drawer-close, ablating a task's top neurons damages another task more than the task itself. Second, the damage is strongly asymmetric. Drawer-close retains 99.7% of its return when its own top neurons are ablated, yet those same neurons reduce window-close to 52.1% and window-open to 62.9%. The reverse does not hold: ablating either window task leaves drawer-close fully intact, at 100.3% and 100.8%. The same one-

Table 4: Seven-task Meta-World subset: off-diagonal BC across the 21 task pairs, and mean Spearman correlation of the pairwise-BC ordering between seeds.

	Off-diagonal BC, mean (range)	Cross-seed ρ of pair ordering
Trained (64 units)	0.775 (0.65–0.90)	0.21
Random (64 units)	0.797 (0.74–0.86)	0.35
Trained (400 units)	0.668 (0.59–0.76)	0.30
Random (400 units)	0.801 (0.74–0.86)	0.68

Table 5: Retained return as % of baseline when ablating the top 5% of neurons ranked by importance for each source task (rows) and evaluating on all tasks (columns) in the seven-task Meta-World subset (400 hidden units). Diagonal entries (bold) reflect self-damage.

Ablated /Eval	reach	door-open	drawer-open	drawer-close	button-press	window-open	window-close
reach	**89.3±14.1**	94.8±28.5	100.5±9.3	99.2±3.7	67.2±34.8	102.4±12.2	91.6±8.7
door-open	90.5±12.2	**49.5±38.8**	91.6±19.9	100.6±2.3	74.7±25.4	92.6±21.5	89.7±12.2
drawer-open	91.6±15.3	63.5±31.7	**93.3±18.5**	100.3±2.3	54.4±36.5	104.4±11.3	85.5±33.0
drawer-close	77.6±34.2	75.9±43.6	88.9±15.5	**99.7±2.6**	70.3±43.1	62.9±48.4	52.1±39.4
button-press	88.5±13.0	83.0±31.0	89.9±21.7	99.8±3.2	**52.7±45.3**	96.0±15.6	79.1±37.5
window-open	87.2±9.5	97.7±27.6	94.4±11.3	100.8±2.2	56.5±37.6	**40.7±36.6**	48.4±44.2
window-close	87.8±17.2	78.6±36.1	76.9±33.6	100.3±2.3	56.9±31.7	64.9±53.5	**47.7±41.5**

directional dependence appears in the original MT10 policies described in Appendix H, so it is not an artifact of incomplete learning. BC, being a symmetric measure, cannot capture this one-directional dependence.

In the seven-task Meta-World subset, moderate importance overlap coexists with weak cross-seed reproducibility of task-pair ordering and asymmetric ablation effects. These results suggest shared computation with unequal functional dependence, which means neurons important to one task can be critical to another without a reciprocal dependence. Together with the locomotion results, this contrast supports task-family geometry as an organizing principle for neuron sharing in the policies studied.

5 Conclusion

Our results support task geometry as an organizing principle for neuron sharing in the PPO-trained MLP policies studied. In AntDir and HalfCheetahVel, nearby tasks have more similar neuron importance profiles than distant tasks. In AntDir this organization survives replacing the continuous goal features with one-hot task IDs, so it is learned from the behavioral structure of the task family. The continuous encoding only sharpens it, most visibly at the smaller width.

A key methodological finding is that importance overlap and functional modularity are dissociable. Both track task geometry in AntDir, but in HalfCheetahVel overlap tracks velocity distance while ablation damage is uniform, consistent with amplitude modulation of broadly shared computation. How similarly two tasks *weight* neurons is therefore distinct from whether they *depend* on distinct neurons, a distinction that matters for interpreting neuron-level analyses and for future work on subnetwork extraction.

Limitations. Our analysis is restricted to MLPs with two hidden layers trained with PPO and to task families with a single, uniformly spaced parameter and symmetric distance; deeper or recurrent policies, other algorithms, and asymmetric or hierarchical task families remain open.

Future directions. The geometry-respecting structure suggests applications in structured pruning, which Appendix E tests as a proof of concept, task-conditioned routing, and transfer from geometric neighbors. The HalfCheetahVel dissociation raises the question of which task properties determine whether a network develops modular or amplitude-scaling solutions.

AI USE STATEMENT

We used generative AI tools for three tasks for which the ICLR 2027 AI Policy requires disclosure: providing feedback on experiments, helping implement the analysis code, and helping interpret results, including the observation in Appendix G. We did not use generative AI tools for any other task with required disclosure. We also used generative AI tools for three tasks with recommended disclosure: creating and refining the data visualizations, drafting parts of the paper, and editing the text for readability. The authors designed the research questions and experiments, reviewed and tested the AI-assisted analysis code, and verified all interpretations against the saved results. Figures are rendered by our analysis pipeline directly from saved experiment outputs. The authors reviewed and revised all AI-drafted and AI-edited text for accuracy. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Section 3.4 describes the environments, observation encodings, policy architecture, training budgets, seeds, and network widths, and Appendix A lists all PPO hyperparameters. Sections 3.1–3.3 define the importance profiles, overlap metrics, and ablation protocol used in all experiments. Section 4.1 describes the random-initialization baselines and the one-hot control, whose training performance is reported in Appendix B. Appendix J reports the compute resources. The supplementary material contains anonymized code for training the policies and for the analysis pipeline used to compute the reported metrics, tables, and figures.

REFERENCES

Jacob Andreas, Dan Klein, and Sergey Levine. Modular multitask reinforcement learning with policy sketches. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 166–175. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/andreas17a.html.

Diana Borsa, Thore Graepel, and John Shawe-Taylor. Learning shared representations in multi-task reinforcement learning. *arXiv (Cornell University)*, February 2016. URL https://arxiv.org/abs/1603.02041.

Tanya Chowdhury, Atharva Nijasure, Yair Zick, and James Allan. Hedonic neurons: A mechanistic mapping of latent coalitions in transformer MLPs. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=v6HPsCu2R8.

Arthur Conmy, Augustine N. Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2304.14997.

Laura N. Driscoll, Krishna Shenoy, Darshan Bhatt, and David Sussillo. Flexible multitask computation in recurrent networks utilizes shared dynamical motifs. *Nature Neuroscience*, 2024. URL https://www.nature.com/articles/s41593-024-01668-6.

Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Vlad Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, Shane Legg, and Koray Kavukcuoglu. IMPALA: Scalable distributed deep-RL with importance weighted actor-learner architectures. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1407–1416. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/espeholt18a.html.

Jenelle Feather and SueYeon Chung. Unveiling the benefits of multitasking in disentangled representation formation. *Trends in Cognitive Sciences*, May 2023. URL https://linkinghub.elsevier.com/retrieve/pii/S1364661323001274.

Samuel Garcin, Trevor McInroe, Pablo Samuel Castro, Christopher G. Lucas, David Abel, Prakash Panangaden, and Stefano V Albrecht. Studying the interplay between the actor and critic representations in reinforcement learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=tErHYBGlWc.

Itamar Hadad, Guy Katz, and Shahaf Bassan. Formal mechanistic interpretability: Automated circuit discovery with provable guarantees. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Timsb74vIY.

Matteo Hessel, Hubert Soyer, Lasse Espeholt, Wojciech Marian Czarnecki, Simon Schmitt, and Hado van Hasselt. Multi-task deep reinforcement learning with popart. In *Proceedings of the AAAI Conference on Artificial Intelligence*, September 2018. URL https://ojs.aaai.org/index.php/AAAI/article/view/4266.

Xinting Huang and Michael Hahn. Decomposing representation space into interpretable subspaces with unsupervised learning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=SI2OZJuvPO.

W. Jeffrey Johnston and Stefano Fusi. Abstract representations emerge naturally in neural networks trained to perform multiple tasks. *Nature Communications*, October 2021. URL https://www.nature.com/articles/s41467-023-36583-0.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2668–2677. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/kim18d.html.

Yi-Ling Liu, Melvin Laux, Mariela De Lucas Alvarez, Frank Kirchner, and Rebecca Adam. Task-specific subnetwork discovery in reinforcement learning for autonomous underwater navigation, 2026. URL https://arxiv.org/abs/2604.21640.

Samuel Marks, Can Rager, Eric J Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=I4e82CIDxv.

Reginald McLean, Evangelos Chatzaroulas, Luc McCutcheon, Frank Röder, Tianhe Yu, Zhanpeng He, K.R. Zentner, Ryan Julian, J K Terry, Isaac Woungang, Nariman Farsad, and Pablo Samuel Castro. Meta-world+: An improved, standardized, RL benchmark. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2025a. URL https://openreview.net/forum?id=1de3azE606.

Reginald McLean, Evangelos Chatzaroulas, J K Terry, Isaac Woungang, Nariman Farsad, and Pablo Samuel Castro. Multi-task reinforcement learning enables parameter scaling. In *Reinforcement Learning Conference*, 2025b. URL https://openreview.net/forum?id=eBWwBIFV7T.

Joshua Brendan Melander, Zaki Alaoui, Shenghua Liu, Surya Ganguli, and Stephen Baccus. Causal interpretation of neural network computations with contribution decomposition. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=6jIc8m38gj.

U. Mini, Peli Grietzer, Mrinank Sharma, Austin Meek, Monte MacDiarmid, and Alexander Turner. Understanding and controlling a maze-solving policy network. *arXiv (Cornell University)*, October 2023. URL https://arxiv.org/abs/2310.08043.

Emilio Parisotto, Jimmy Ba, and Ruslan Salakhutdinov. Actor-mimic: Deep multitask and transfer reinforcement learning. In *International Conference on Learning Representations*, 2016. URL https://arxiv.org/abs/1511.06342.

Clemens M. Rosenbaum, Tim Klinger, and Matthew Riemer. Routing networks: Adaptive selection of non-linear functions for multi-task learning. *arXiv (Cornell University)*, November 2017. URL https://arxiv.org/abs/1711.01239.

John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In Yoshua Bengio and Yann LeCun (eds.), *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL http://arxiv.org/abs/1506.02438.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

Aaquib Syed, Can Rager, and Arthur Conmy. Attribution patching outperforms automated circuit discovery. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2310.10348.

Mark Towers, Ariel Kwiatkowski, John Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Kallinteris Andreas, Markus Krimmel, Arjun KG, Rodrigo Perez-Vicente, J Terry, Andrea Pierré, Sander Schulhoff, Jun Jet Tai, Hannah Tan, and Omar G. Younis. Gymnasium: A standard interface for reinforcement learning environments. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38. Curran Associates, Inc., 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/d7ff1795e8527f6443371c3933bdb52b-Paper-Datasets_and_Benchmarks_Track.pdf.

Tristan Trim and Triston Grayston. Mechanistic interpretability of reinforcement learning agents. *arXiv (Cornell University)*, October 2024. URL https://arxiv.org/abs/2411.00867.

Pantelis Vafidis, Aman Bhargava, and Antonio Rangel. Disentangling representations through multi-task learning. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 68296–68338, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/ab1b7c99395064743acbc6c694e4253a-Paper-Conference.pdf.

Paweł Wawrzyński. A cat-like robot real-time learning to run. In *Proceedings of the 2009 Conference on Adaptive and Natural Computing Algorithms - Volume 5495*, ICANNGA 2009, pp. 380–390, Berlin, Heidelberg, 2009. Springer-Verlag. ISBN 9783642049200. doi: 10.1007/978-3-642-04921-7_39. URL https://doi.org/10.1007/978-3-642-04921-7_39.

A PPO HYPERPARAMETERS

Table 6 lists the PPO hyperparameters used for each environment. All environments use the same base algorithm with environment-specific tuning. The network architecture consists of separate actor and critic MLPs, each with two hidden layers of equal width and ReLU activations. Weights are initialized with orthogonal initialization ($\sqrt{2}$ gain for hidden layers, 0.01 for the actor output, 1.0 for the critic output). The action distribution is a diagonal Gaussian with learned per-dimension log-standard-deviation parameters. The exact five seeds we used in this experiment were 1000, 1001, 1002, 1003, and 1004. We used the same seeds across all environments and architectures to ensure comparability.

Table 6: PPO hyperparameters for each environment.

Hyperparameter	AntDir	HalfCheetahVel	Meta-World MT10
Total timesteps	50M	50M	20M
Num. parallel envs	32	32	10
Steps per rollout	1024	1024	960
Learning rate	3e-4	3e-4	3e-4
LR annealing	No	No	No
Discount (γ)	0.99	0.99	0.99
GAE λ	0.95	0.95	0.97
Clip coefficient	0.2	0.2	0.2
Entropy coefficient	0.005	0.005	0.005
Value function coeff.	0.5	0.5	0.5
Max gradient norm	0.5	0.5	1.0
Update epochs	10	10	16
Num. minibatches	32	32	32
Normalize advantages	Yes	Yes	No
Clip value loss	Yes	Yes	No
Hidden layer width	64 / 400	64	64 / 400
Activation	ReLU	ReLU	ReLU

B ONE-HOT CONTROL: TRAINING PERFORMANCE

The one-hot control policies of Section 4.1 are trained with hyperparameters, seeds, direction pool, and training budget identical to the continuous-encoding policies; only the goal encoding differs. Table 7 compares evaluation returns. One-hot policies reach return parity with their continuous counterparts at both widths, so the structural comparisons in the main text are not confounded by training quality.

Table 7: Grand-mean evaluation return across the 8 directions and 5 seeds for the continuous and one-hot AntDir policies. Random-policy return is $\approx$995 in all conditions.

Width	Continuous	One-hot	Parity
64	4464	4359	98%
400	2958	2938	99%

C SEVEN-TASK META-WORLD SUBSET: TRAINING PERFORMANCE

The seven-task policies of Section 4.3 are trained with the architecture, hyperparameters, and training budget of the MT10 policies. Table 8 shows that every task clears its random baseline at both widths.

Table 8: Mean episodic return per task in the seven-task Meta-World subset for trained policies and random-initialization baselines, mean ± std across 5 seeds.

	64 units		400 units	
Task	Trained	Random	Trained	Random
reach	4587 ± 252	738 ± 101	4211 ± 335	721 ± 9
door-open	3841 ± 663	294 ± 16	3711 ± 827	283 ± 6
drawer-open	2895 ± 798	616 ± 52	2393 ± 206	605 ± 8
drawer-close	4755 ± 162	499 ± 1034	4815 ± 101	141 ± 304
button-press-topdown	2560 ± 251	64 ± 42	2825 ± 618	24 ± 31
window-open	2633 ± 1385	207 ± 3	3809 ± 356	205 ± 0.3
window-close	4246 ± 201	264 ± 13	4153 ± 222	259 ± 4

D Ablation Results at Additional Sparsity Levels

The main text reports ablation results at $s = 0.05$ (removing the top 5% of neurons), summarized by angular distance for AntDir. Here we provide the full per-task matrices at $s = 0.05$ and results for additional sparsity levels ($s \in \{0.10, 0.20, 0.30, 0.50\}$) to show that the qualitative patterns are consistent.

For AntDir, increasing sparsity uniformly increases both self-damage and collateral damage, but the angular gradient in collateral damage persists at all levels. At $s = 0.50$, most tasks are near-completely degraded, but the ordering of damage by angular distance remains visible. The monotonic relationship between sparsity and damage confirms that the importance ranking is well-calibrated: neurons ranked higher are indeed more critical.

For HalfCheetahVel, the pattern of uniform row-wise damage persists at all sparsity levels. Increasing sparsity amplifies damage across the board but does not introduce any diagonal penalty, confirming that the lack of functional modularity is not an artifact of the 5% threshold. The cells in the table that have no values indicate that return was negative not allowing us to compute a percentage of retained return.

For Meta-World MT10, increasing sparsity amplifies the existing asymmetries without introducing new structure. The ad-hoc, task-dependent pattern of collateral damage remains at all levels.

Table 9: Retained return (% of baseline) after ablating the top 5% of neurons for AntDir 8-dir (64 units).

Ablated / Eval	NE	NW	SW	SE	N	W	S	E
NE	**34.1±19.7**	87.8±27.5	73.5±36.4	75.7±35.6	51.9±29.0	74.8±36.7	79.8±34.5	61.3±32.8
NW	86.5±10.6	**47.9±32.5**	64.3±34.6	67.8±41.5	68.3±30.4	55.4±28.1	72.4±34.7	81.8±38.8
SW	69.1±40.3	56.5±37.1	**41.5±19.5**	49.6±24.3	66.5±39.9	51.0±27.9	50.1±31.6	60.7±42.7
SE	82.5±28.4	62.4±50.9	54.0±35.9	**24.7±9.3**	72.6±36.2	62.4±42.4	24.8±23.8	50.2±30.3
N	61.4±32.3	59.5±43.4	70.5±34.4	91.7±8.9	**45.7±38.7**	61.5±36.6	76.0±40.5	82.2±32.8
W	63.3±29.2	41.2±27.0	63.5±29.3	66.7±33.3	45.0±30.3	**33.7±28.2**	63.7±34.5	71.0±40.2
S	78.4±37.0	66.0±41.3	46.9±30.3	32.0±22.4	74.5±40.2	64.1±33.0	**31.6±22.1**	55.6±33.5
E	74.9±30.2	78.9±46.9	75.4±32.1	56.5±38.6	92.3±17.8	79.8±33.0	52.1±41.7	**45.5±27.0**

Table 10: Retained return (% of baseline) after ablating the top 10% of neurons for AntDir 8-dir (64 units).

Ablated / Eval	NE	NW	SW	SE	N	W	S	E
NE	**39.5±9.3**	59.5±41.6	52.2±41.4	76.1±34.7	50.0±27.0	54.0±41.8	58.3±44.7	62.6±28.8
NW	44.7±18.4	**31.8±20.5**	52.1±16.4	44.9±45.3	30.4±18.9	34.6±19.8	55.0±38.0	47.8±29.1
SW	62.9±34.3	57.4±36.6	**44.0±22.0**	47.0±23.4	56.5±42.6	49.7±24.7	53.2±30.9	53.7±39.6
SE	69.1±25.1	64.6±39.9	56.8±24.1	**20.1±4.6**	67.8±35.9	64.4±32.3	31.9±22.5	47.0±29.5
N	44.4±13.7	35.1±31.8	44.6±37.8	71.0±34.3	**36.7±26.6**	29.2±24.1	49.0±40.6	67.3±24.8
W	50.0±35.6	32.3±18.3	42.0±28.2	51.7±30.2	43.9±30.3	**24.8±16.7**	56.6±27.1	50.3±44.2
S	68.5±36.3	60.9±39.8	54.2±28.9	23.2±9.2	71.5±43.8	61.3±32.7	**28.4±20.6**	39.4±30.3
E	57.0±36.9	72.2±44.8	73.6±30.8	46.5±37.1	72.5±36.8	77.7±30.8	48.8±37.8	**24.7±10.7**

Table 11: Retained return (% of baseline) after ablating the top 20% of neurons for AntDir 8-dir (64 units).

Ablated / Eval	NE	NW	SW	SE	N	W	S	E
NE	**18.7±7.4**	31.8±34.7	20.3±26.6	22.4±14.3	33.8±25.8	26.3±27.3	6.7±4.6	37.4±24.1
NW	28.7±8.1	**33.8±21.0**	16.2±12.8	15.3±25.2	27.1±17.9	31.7±15.9	17.4±19.2	20.3±14.7
SW	41.1±34.1	25.9±19.5	**23.1±10.3**	31.6±16.4	21.6±19.0	22.9±13.5	34.8±25.7	37.4±32.1
SE	44.7±32.3	60.7±34.6	38.4±33.6	**20.0±8.0**	66.8±23.7	53.8±36.2	23.1±6.6	27.8±22.5
N	20.4±3.6	29.6±22.2	15.0±12.4	26.7±25.1	**27.3±14.9**	20.6±14.2	11.1±8.9	27.5±20.0
W	23.2±19.1	31.8±25.2	22.8±10.4	38.5±29.7	31.1±24.6	**22.7±15.1**	38.0±14.8	15.3±14.5
S	54.8±25.6	51.5±30.0	43.9±27.9	22.9±14.8	71.3±20.3	52.2±24.2	**21.8±9.2**	39.3±29.1
E	28.7±34.0	46.5±44.8	33.6±35.2	32.2±24.5	38.4±39.8	40.1±38.5	25.9±27.5	**18.8±7.5**

Table 12: Retained return (% of baseline) after ablating the top 30% of neurons for AntDir 8-dir (64 units).

Ablated / Eval	NE	NW	SW	SE	N	W	S	E
NE	**18.9±5.7**	9.6±9.6	10.3±9.2	13.2±11.7	16.3±7.4	8.4±8.8	6.8±3.8	17.0±9.0
NW	26.9±11.2	**27.1±14.9**	15.4±11.4	10.6±8.8	19.9±16.4	27.0±11.0	15.8±18.3	21.9±20.0
SW	33.7±30.2	26.6±12.9	**23.6±3.9**	16.3±10.3	24.9±22.3	20.0±6.0	24.9±8.5	14.5±13.7
SE	20.6±12.8	34.8±33.8	24.2±9.1	**22.7±6.1**	19.1±25.5	25.7±30.5	21.0±5.8	34.4±23.0
N	21.7±9.1	19.0±16.4	11.2±9.3	15.5±12.4	**20.7±5.1**	14.3±11.0	11.8±13.0	22.1±17.6
W	16.4±14.1	26.6±12.6	18.6±6.3	20.4±21.3	21.2±15.1	**19.8±6.3**	22.1±10.9	14.5±17.9
S	39.2±29.6	31.2±31.9	26.8±14.9	28.1±17.4	34.5±27.1	32.0±20.1	**21.4±6.0**	40.5±15.9
E	24.6±31.0	24.6±33.5	23.1±18.7	20.6±11.6	21.3±28.7	22.2±31.0	19.8±17.9	**22.1±7.7**

Table 13: Retained return (% of baseline) after ablating the top 50% of neurons for AntDir 8-dir (64 units).

Ablated / Eval	NE	NW	SW	SE	N	W	S	E
NE	**19.9±1.9**	14.3±9.0	8.6±8.5	6.9±6.4	18.6±1.4	10.4±7.2	4.4±3.1	19.9±8.8
NW	14.1±7.5	**22.5±6.2**	16.3±8.8	19.2±14.3	18.1±9.0	19.4±2.9	17.7±10.6	24.6±18.9
SW	16.5±11.7	25.2±8.4	**17.6±3.4**	17.4±8.7	14.0±9.6	19.2±3.2	19.8±5.5	17.1±15.4
SE	18.1±13.4	19.9±20.1	25.8±15.4	**20.6±2.2**	16.2±17.4	19.7±12.9	18.9±2.3	20.3±14.5
N	18.2±3.6	21.6±5.5	11.5±6.7	14.8±11.0	**19.1±2.0**	18.5±2.6	14.3±11.6	20.6±14.1
W	9.6±6.3	23.1±6.6	15.6±4.0	24.4±10.6	12.0±7.9	**18.1±2.4**	17.7±9.1	19.6±22.1
S	18.1±12.1	21.1±11.0	23.1±8.4	19.1±3.5	13.4±9.8	18.5±4.9	**19.1±2.2**	25.6±15.9
E	18.6±8.3	17.6±16.3	12.1±5.1	16.6±8.5	19.2±14.4	9.2±9.8	12.1±7.5	**20.9±6.1**

Table 14: Retained return (% of baseline) after ablating the top 5% of neurons for AntDir 8-dir (400 units).

Ablated / Eval	NE	NW	SW	SE	N	W	S	E
NE	**50.2±17.8**	89.9±23.8	93.8±52.2	88.0±34.7	83.5±35.1	82.9±31.2	92.9±48.3	62.1±45.9
NW	101.3±41.9	**52.1±15.6**	84.7±40.8	99.3±42.8	80.2±30.9	55.6±24.4	80.9±36.9	102.7±67.8
SW	97.5±41.0	92.3±27.1	**52.8±18.8**	94.6±46.4	98.9±35.8	77.1±35.4	60.0±22.9	97.7±67.9
SE	79.3±38.9	96.8±22.0	94.0±48.3	**56.3±24.6**	101.9±38.0	99.4±19.6	99.3±40.4	51.6±29.5
N	70.6±41.5	67.6±28.2	82.8±54.2	86.9±37.4	**48.0±17.1**	72.9±23.1	87.3±47.0	83.5±58.7
W	86.4±34.4	60.4±18.7	61.5±27.6	95.8±47.9	70.4±23.1	**46.2±17.8**	76.8±32.8	99.6±67.0
S	87.3±34.8	94.6±25.8	54.5±20.9	88.0±43.8	101.3±34.8	79.7±39.7	**55.5±17.5**	73.7±58.3
E	76.9±37.8	83.0±32.5	70.4±49.7	66.4±26.8	98.6±31.8	82.0±29.4	74.9±42.2	**50.4±26.8**

Table 15: Retained return (% of baseline) after ablating the top 10% of neurons for AntDir 8-dir (400 units).

Ablated / Eval	NE	NW	SW	SE	N	W	S	E
NE	**37.9±12.5**	76.1±38.8	88.0±47.7	76.0±26.8	59.2±22.3	67.1±38.2	83.2±38.4	57.5±39.8
NW	102.2±38.0	**44.7±12.4**	72.0±38.9	95.8±47.9	77.3±34.6	38.1±10.4	77.7±34.9	98.3±67.0
SW	85.4±37.4	83.2±23.1	**48.4±17.1**	94.1±44.3	90.9±34.9	70.5±28.0	60.7±21.7	73.4±62.1
SE	78.1±37.8	71.3±32.7	75.9±34.6	**51.8±24.0**	87.4±31.9	70.3±39.5	71.7±33.0	50.6±27.3
N	62.0±43.1	58.5±21.8	77.3±47.7	85.4±37.1	**38.6±11.4**	58.7±17.2	80.8±40.6	82.2±57.3
W	81.0±36.3	57.0±13.3	66.5±33.2	98.3±49.3	64.5±20.2	**41.7±8.2**	76.3±33.4	97.9±64.7
S	84.3±35.2	89.5±29.5	49.0±17.4	84.8±46.0	97.9±37.0	79.4±38.5	**48.5±16.4**	69.0±52.3
E	76.2±34.9	56.5±26.0	72.4±41.7	56.2±25.2	88.0±33.1	56.8±30.2	81.1±34.9	**42.2±21.8**

Table 16: Retained return (% of baseline) after ablating the top 20% of neurons for AntDir 8-dir (400 units).

Ablated / Eval	NE	NW	SW	SE	N	W	S	E
NE	**34.5±12.4**	65.8±32.7	85.6±49.4	82.8±30.5	55.6±22.3	54.6±34.8	75.0±42.7	63.4±40.0
NW	85.6±29.9	**37.2±8.9**	67.9±37.8	76.1±43.0	63.8±27.8	38.3±7.0	76.1±36.3	78.8±59.3
SW	57.7±30.5	54.8±12.4	**38.8±14.9**	65.5±33.0	65.5±19.8	41.6±15.5	51.5±18.6	54.1±42.9
SE	44.9±24.6	35.9±26.2	58.6±40.6	**41.1±16.8**	41.5±30.7	31.8±20.4	58.2±25.9	45.5±23.1
N	58.3±42.5	40.6±8.7	69.5±48.8	85.9±37.3	**33.9±11.3**	46.4±14.4	66.9±38.2	85.7±54.3
W	81.4±38.9	48.1±10.1	56.8±29.1	72.2±31.4	57.1±21.3	**36.8±7.2**	66.0±32.5	71.2±55.9
S	74.2±38.0	58.1±32.6	43.1±19.5	69.6±29.3	69.5±47.0	63.1±33.1	**46.3±14.5**	59.4±43.0
E	55.9±21.7	48.4±23.5	64.8±45.6	49.9±21.3	64.5±30.5	44.9±33.1	72.1±40.1	**37.6±19.1**

Table 17: Retained return (% of baseline) after ablating the top 30% of neurons for AntDir 8-dir (400 units).

Ablated / Eval	NE	NW	SW	SE	N	W	S	E
NE	**32.6±10.0**	58.8±29.9	70.9±40.2	62.1±18.9	42.0±11.8	64.2±20.5	77.5±33.3	53.1±28.1
NW	67.0±26.9	**32.2±6.1**	59.5±34.8	79.6±40.5	48.4±18.9	38.4±8.0	73.4±36.5	63.1±38.9
SW	66.0±38.0	54.1±16.4	**34.6±12.8**	65.8±39.8	64.8±21.3	34.7±10.8	44.0±18.3	52.8±40.4
SE	43.9±23.4	29.6±25.6	47.8±30.7	**36.4±11.5**	33.3±29.7	36.1±17.1	42.7±14.8	46.5±23.6
N	47.6±26.4	40.2±14.2	70.0±43.2	80.4±31.7	**33.7±10.0**	51.4±22.2	69.8±36.1	63.1±46.7
W	78.2±37.4	44.4±12.7	49.2±23.7	75.8±34.0	52.4±21.1	**34.8±6.2**	64.6±25.6	74.8±56.8
S	38.4±23.4	46.7±27.5	42.6±15.5	44.6±18.4	35.0±32.5	56.4±24.9	**35.3±10.5**	49.8±27.8
E	47.1±15.8	48.3±22.3	44.0±41.6	48.6±15.5	61.2±30.3	38.9±31.1	64.3±34.5	**35.4±17.2**

Table 18: Retained return (% of baseline) after ablating the top 50% of neurons for AntDir 8-dir (400 units).

Ablated / Eval	NE	NW	SW	SE	N	W	S	E
NE	**31.2±8.7**	44.9±21.8	43.7±30.0	44.4±14.2	34.1±10.0	41.3±8.5	59.6±23.9	36.0±16.9
NW	55.4±25.7	**28.7±4.4**	38.9±17.1	66.3±24.8	36.0±10.3	28.3±4.5	50.0±19.9	54.2±29.9
SW	52.1±23.5	38.1±12.5	**31.6±11.1**	46.1±19.4	56.0±21.2	28.6±4.5	35.0±10.7	52.9±35.7
SE	56.8±22.5	27.1±17.3	49.8±24.3	**35.7±10.4**	44.4±30.3	37.6±13.6	34.8±10.2	36.0±17.4
N	36.3±11.1	28.4±4.7	49.1±24.9	52.9±18.5	**31.2±8.0**	34.9±10.0	56.0±22.8	45.5±25.8
W	59.7±38.3	32.4±6.9	33.7±11.8	46.8±25.1	53.0±22.7	**27.3±4.1**	54.3±22.8	46.6±48.9
S	36.6±19.5	34.5±22.1	33.7±11.9	36.0±11.2	28.1±19.1	35.2±7.7	**33.0±9.8**	40.9±20.2
E	35.0±9.9	46.0±23.7	44.3±27.9	35.8±11.0	48.2±20.0	39.4±20.2	51.9±19.9	**33.9±15.9**

Table 19: Retained return (% of baseline) after ablating the top 5% of neurons for HalfCheetahVel (64 units).

Ablated / Eval	0.25	0.50	0.75	1.00	1.25	1.50	1.75	2.00
0.25	**60.8±37.8**	51.9±44.7	51.4±44.1	45.2±55.2	41.2±60.1	3.3±127.4	–	–
0.50	66.0±22.5	**45.4±44.3**	44.2±36.9	11.0±70.0	–	–	–	–
0.75	66.7±20.5	49.6±31.1	**35.3±33.0**	21.0±41.5	–	–	–	–
1.00	68.0±19.5	48.4±35.1	30.2±37.9	**7.4±51.0**	–	–	–	–
1.25	67.8±19.4	47.2±34.9	24.2±35.6	–	–	–	–	–
1.50	67.4±19.2	44.0±36.2	21.8±36.0	–	–	–	–	–
1.75	59.8±35.6	43.3±36.8	18.4±37.5	–	–	–	–	–
2.00	58.1±35.2	43.5±36.8	19.8±37.8	–	–	–	–	–

Table 20: Retained return (% of baseline) after ablating the top 10% of neurons for HalfCheetahVel (64 units).

Ablated / Eval	0.25	0.50	0.75	1.00	1.25	1.50	1.75	2.00
0.25	**63.1±30.5**	55.5±34.2	41.7±52.0	36.1±47.3	–	–	–	–
0.50	–	–	–	–	–	–	–	–
0.75	5.0±121.5	–	–	–	–	–	–	–
1.00	2.3±127.4	–	–	–	–	–	–	–
1.25	1.3±127.8	–	–	–	–	–	–	–
1.50	2.5±128.8	–	–	–	–	–	–	–
1.75	12.2±127.3	–	–	–	–	–	–	–
2.00	12.1±127.1	–	–	–	–	–	–	–

Table 21: Retained return (% of baseline) after ablating the top 20% of neurons for HalfCheetahVel (64 units).

Ablated / Eval	0.25	0.50	0.75	1.00	1.25	1.50	1.75	2.00
0.25	**43.8±51.6**	23.9±45.3	3.2±60.2	–	–	–	–	–
0.50	42.5±40.8	–	–	–	–	–	–	–
0.75	41.3±48.5	–	–	–	–	–	–	–
1.00	61.7±12.4	–	–	–	–	–	–	–
1.25	61.5±17.9	–	–	–	–	–	–	–
1.50	46.4±46.9	–	–	–	–	–	–	–
1.75	65.6±17.5	–	–	–	–	–	–	–
2.00	63.7±19.9	–	–	–	–	–	–	–

Table 22: Retained return (% of baseline) after ablating the top 30% of neurons for HalfCheetahVel (64 units).

Ablated / Eval	0.25	0.50	0.75	1.00	1.25	1.50	1.75	2.00
0.25	**22.1±44.0**	–	–	–	–	–	–	–
0.50	33.5±41.2	–	–	–	–	–	–	–
0.75	57.8±14.6	–	–	–	–	–	–	–
1.00	59.5±15.8	–	–	–	–	–	–	–
1.25	13.8±59.7	–	–	–	–	–	–	–
1.50	21.3±53.2	–	–	–	–	–	–	–
1.75	3.0±68.0	–	–	–	–	–	–	–
2.00	20.4±46.4	–	–	–	–	–	–	–

Table 23: Retained return (% of baseline) after ablating the top 50% of neurons for HalfCheetahVel (64 units).

Ablated / Eval	0.25	0.50	0.75	1.00	1.25	1.50	1.75	2.00
0.25	**34.8±42.8**	–	–	–	–	–	–	–
0.50	30.0±49.4	–	–	–	–	–	–	–
0.75	43.7±21.7	–	–	–	–	–	–	–
1.00	23.5±64.2	–	–	–	–	–	–	–
1.25	10.5±75.6	–	–	–	–	–	–	–
1.50	49.1±17.8	–	–	–	–	–	–	–
1.75	57.1±9.2	–	–	–	–	–	–	–
2.00	44.4±15.9	–	–	–	–	–	–	–

Table 24: Retained return (% of baseline) after ablating the top 5% of neurons for Meta-World MT10 (64 units).

Ablated / Eval	btn-press	door-open	drawer-cl	drawer-op	peg-ins	pick-pl	push	reach	win-close	win-open
btn-press	**45.9±40.1**	88.9±46.3	94.9±7.9	78.7±16.1	118.0±208.1	95.5±62.0	57.3±72.9	82.8±49.4	55.4±45.8	87.0±27.1
door-open	66.0±26.0	**47.1±22.9**	98.0±5.0	99.2±3.2	279.1±613.4	103.0±48.1	35.6±45.3	84.5±43.2	99.2±77.1	85.4±40.4
drawer-cl	80.9±44.1	85.4±47.3	**96.2±5.0**	94.4±12.7	309.5±683.3	104.3±67.1	73.5±97.0	104.3±34.9	83.5±75.2	70.7±32.3
drawer-op	68.3±33.5	102.1±48.3	100.1±2.2	**65.5±23.2**	275.9±603.6	120.0±102.0	40.0±75.2	76.6±48.7	61.3±63.1	61.9±41.4
peg-ins	93.5±37.2	96.4±39.2	96.3±10.1	98.9±3.3	**26.0±29.2**	89.6±43.3	101.4±117.0	96.6±39.5	99.0±77.2	102.5±33.0
pick-pl	84.7±33.8	93.7±41.9	96.5±7.9	93.2±13.0	38.4±46.9	**451.8±799.5**	107.8±154.4	108.6±34.4	94.3±72.5	95.8±34.8
push	72.2±52.2	74.2±49.2	96.9±7.7	98.4±8.4	43.6±51.2	105.7±53.6	**47.4±59.7**	95.6±39.8	95.4±77.2	48.9±26.3
reach	68.1±27.2	62.6±37.5	96.9±9.4	87.4±15.5	57.5±72.2	96.0±55.9	41.8±51.3	**60.7±51.6**	75.3±60.5	45.3±29.2
win-close	68.5±27.9	80.9±35.7	101.0±2.2	94.2±7.9	239.9±527.4	93.9±56.0	47.7±56.7	73.3±49.8	**68.9±58.7**	70.2±44.3
win-open	60.8±27.6	79.5±45.0	99.7±2.7	66.9±23.4	125.3±232.0	93.3±58.0	54.5±71.1	65.2±49.5	59.8±52.6	**40.8±29.0**

Table 25: Retained return (% of baseline) after ablating the top 10% of neurons for Meta-World MT10 (64 units).

Ablated / Eval	btn-press	door-open	drawer-cl	drawer-op	peg-ins	pick-pl	push	reach	win-close	win-open
btn-press	**31.7±31.2**	56.4±38.7	101.2±2.3	82.3±17.1	49.4±61.3	92.2±60.5	37.4±46.5	88.5±52.9	53.2±41.8	66.9±40.4
door-open	58.8±25.7	**45.9±38.3**	100.8±2.3	92.1±9.2	103.3±178.0	106.2±55.7	19.7±22.6	77.7±34.4	87.9±71.7	54.1±46.4
drawer-cl	74.8±38.3	63.4±39.0	**90.0±17.9**	84.9±15.1	252.7±566.4	110.4±64.7	64.5±88.1	90.9±29.3	77.4±70.3	34.3±24.3
drawer-op	54.6±29.7	61.3±52.4	91.3±17.5	**76.6±15.8**	261.2±585.4	111.9±64.6	39.6±61.1	72.5±45.5	52.8±54.7	34.2±33.4
peg-ins	82.7±27.1	93.7±43.4	89.7±17.1	97.8±3.8	**55.0±75.6**	104.2±68.6	115.2±163.0	90.8±41.6	71.4±71.8	101.0±35.1
pick-pl	69.8±39.9	64.1±48.0	98.4±5.0	76.3±25.5	33.0±41.0	**75.4±58.8**	48.6±87.6	87.7±47.1	74.1±62.3	58.0±51.1
push	55.0±43.6	54.0±34.0	98.4±5.3	98.5±4.3	42.5±48.6	104.5±56.0	**33.6±43.3**	81.3±49.2	90.8±75.5	46.5±27.4
reach	61.9±49.6	55.8±37.0	99.6±3.5	83.1±22.1	55.0±69.9	385.8±673.0	37.1±45.5	**53.5±44.4**	39.5±58.1	51.6±21.2
win-close	65.4±31.2	66.5±30.0	100.4±2.7	96.2±6.4	267.6±605.8	224.6±313.4	84.0±123.5	69.0±31.9	**80.2±75.4**	72.0±42.2
win-open	53.2±40.4	74.8±40.4	99.6±2.2	50.2±27.7	143.1±279.2	92.1±54.7	44.0±63.4	41.5±35.2	68.0±60.8	**25.9±26.2**

Table 26: Retained return (% of baseline) after ablating the top 20% of neurons for Meta-World MT10 (64 units).

Ablated / Eval	btn-press	door-open	drawer-cl	drawer-op	peg-ins	pick-pl	push	reach	win-close	win-open
btn-press	**10.8±10.9**	31.3±22.7	100.5±2.6	61.6±15.8	68.1±97.3	106.6±70.7	42.9±72.8	67.1±42.3	54.3±41.9	43.9±29.6
door-open	47.6±28.4	**36.0±30.1**	82.6±30.9	65.4±30.3	60.5±98.1	75.5±51.8	35.5±49.7	64.2±39.3	76.3±61.0	40.0±38.9
drawer-cl	56.1±29.8	41.5±38.4	**86.0±30.2**	64.4±22.7	105.0±194.6	104.2±58.5	39.4±47.6	63.0±36.9	70.3±69.0	41.2±49.4
drawer-op	40.2±31.1	58.1±44.6	80.7±17.2	**51.1±34.0**	85.8±166.0	76.4±43.2	39.3±58.2	55.2±44.6	60.8±70.2	14.6±17.2
peg-ins	74.5±28.0	80.5±48.1	98.3±4.0	86.4±9.2	**33.7±40.0**	84.6±39.9	52.6±64.2	80.3±30.3	64.2±64.7	60.4±40.8
pick-pl	64.7±40.4	51.7±46.9	100.3±2.7	74.8±26.8	41.7±58.3	**71.3±41.4**	42.0±79.1	69.6±33.8	70.1±70.2	71.8±33.3
push	70.4±45.6	69.6±37.8	95.7±7.0	91.9±14.1	29.7±33.9	795.3±1621.7	**55.9±92.8**	87.6±32.2	71.3±66.3	52.7±36.2
reach	45.1±37.7	35.2±34.0	96.6±9.2	85.5±22.3	38.7±46.1	116.2±65.0	26.6±36.3	**63.3±35.0**	41.3±51.6	55.4±45.2
win-close	44.1±34.0	46.1±36.2	84.4±25.3	73.1±28.0	41.4±50.2	93.8±53.2	103.1±178.2	74.9±37.3	**75.9±72.0**	34.0±33.7
win-open	34.2±20.5	29.2±23.4	96.3±6.5	62.8±19.0	36.6±42.0	62.8±43.3	66.6±91.4	28.6±27.7	66.9±61.6	**8.0±7.2**

Table 27: Retained return (% of baseline) after ablating the top 30% of neurons for Meta-World MT10 (64 units).

Ablated / Eval	btn-press	door-open	drawer-cl	drawer-op	peg-ins	pick-pl	push	reach	win-close	win-open
btn-press	**17.9±22.2**	40.3±30.0	98.9±5.1	43.3±28.3	35.8±44.5	166.6±149.9	41.3±60.2	56.0±50.1	82.1±65.4	25.4±27.2
door-open	15.5±24.7	**16.6±6.8**	85.9±16.5	54.0±28.4	40.6±48.2	99.9±86.0	25.5±33.9	58.7±43.4	43.5±55.1	28.1±32.6
drawer-cl	50.0±36.9	28.0±34.1	**87.0±25.6**	46.8±30.7	53.6±77.3	69.3±36.0	26.6±43.6	77.3±47.2	46.5±56.4	18.1±19.8
drawer-op	15.3±11.2	32.6±23.7	78.2±43.8	**33.3±12.4**	38.6±54.4	61.5±41.1	21.0±36.0	55.8±49.2	32.1±53.8	18.9±22.8
peg-ins	69.8±24.9	39.8±41.7	97.1±5.5	63.4±24.9	**34.4±42.8**	64.9±39.3	18.9±23.9	54.8±44.3	26.0±30.7	53.0±41.7
pick-pl	62.5±35.5	46.4±39.1	97.2±6.9	66.9±38.7	36.5±50.0	**100.7±52.6**	15.8±22.0	61.4±30.1	63.5±59.1	63.6±51.6
push	55.3±36.2	67.8±43.9	88.7±20.0	65.2±17.3	32.7±37.8	194.7±262.8	**22.9±29.2**	45.5±40.0	83.2±74.7	31.6±30.6
reach	38.8±33.1	38.1±28.4	99.6±4.2	71.0±29.4	52.1±74.5	84.7±45.1	18.1±22.3	**50.0±19.5**	60.0±39.2	45.4±28.7
win-close	32.0±31.9	47.3±36.7	82.0±24.1	61.8±22.7	225.2±493.8	144.7±109.1	24.9±40.7	45.2±33.9	**42.6±58.7**	22.9±26.6
win-open	34.8±19.4	25.3±12.7	89.0±13.3	63.5±27.1	56.9±75.9	92.7±70.3	24.0±31.6	25.9±32.2	58.7±46.5	**7.5±4.2**

Table 28: Retained return (% of baseline) after ablating the top 50% of neurons for Meta-World MT10 (64 units).

Ablated / Eval	btn-press	door-open	drawer-cl	drawer-op	peg-ins	pick-pl	push	reach	win-close	win-open
btn-press	**14.5±13.4**	16.5±10.1	97.5±5.5	30.7±21.6	24.4±28.6	90.8±56.2	21.3±26.3	46.1±41.0	44.9±51.4	14.0±18.5
door-open	9.6±15.5	**8.4±6.4**	47.0±33.4	20.7±10.4	30.6±38.4	398.6±702.2	37.5±59.9	31.5±28.5	26.0±50.6	17.3±28.7
drawer-cl	20.9±19.9	34.2±55.7	**63.5±28.6**	38.9±25.3	22.7±26.7	52.5±30.6	10.2±12.4	47.3±38.8	42.5±55.1	3.8±1.1
drawer-op	14.8±24.6	14.5±11.9	78.3±43.8	**53.7±13.6**	14.9±16.6	319.4±645.6	13.8±29.4	33.2±43.6	8.8±6.7	9.2±10.9
peg-ins	59.8±22.9	36.5±38.4	98.4±5.1	64.5±27.6	**33.5±36.8**	176.4±234.6	49.5±71.0	63.9±38.8	44.8±60.8	31.4±34.2
pick-pl	24.3±26.3	26.3±27.4	97.0±5.5	39.5±31.7	34.1±54.0	**67.0±35.8**	6.5±8.8	38.4±39.4	35.0±52.0	30.1±27.9
push	26.7±25.1	37.0±31.3	91.0±21.6	44.0±17.7	30.5±37.8	218.2±255.6	**18.4±23.9**	38.4±43.8	44.4±55.8	23.2±28.0
reach	19.9±22.7	40.0±36.6	95.3±4.3	37.5±22.2	32.4±39.6	62.7±40.7	9.6±11.6	**44.5±31.8**	26.9±42.2	9.5±6.5
win-close	40.1±34.8	19.4±20.1	77.0±43.2	55.2±37.6	34.0±37.4	96.6±49.4	25.4±47.0	42.3±35.5	**17.4±18.6**	12.2±9.9
win-open	13.8±11.2	30.5±16.6	89.0±24.7	40.2±26.9	23.5±28.1	85.3±59.8	11.3±18.1	30.4±18.5	23.2±27.5	**8.6±2.8**

Table 29: Retained return (% of baseline) after ablating the top 5% of neurons for Meta-World MT10 (400 units).

Ablated / Eval	btn-press	door-open	drawer-cl	drawer-op	peg-ins	pick-pl	push	reach	win-close	win-open
btn-press	**80.4±35.9**	56.5±41.9	99.5±7.8	88.2±16.5	100.9±83.5	96.0±75.3	207.9±322.5	93.8±6.6	80.8±37.9	83.3±23.7
door-open	63.0±34.5	**67.5±54.0**	100.3±8.5	92.0±11.5	107.1±89.1	118.9±109.1	93.9±134.8	93.8±8.3	63.5±47.0	83.8±24.3
drawer-cl	84.9±26.9	76.8±51.5	**98.9±7.0**	81.4±27.4	82.8±59.2	116.8±94.2	170.7±306.6	94.8±7.8	51.3±54.1	31.4±30.9
drawer-op	84.7±31.0	54.5±31.4	101.2±7.2	**81.6±24.4**	104.3±97.1	79.4±51.5	97.8±129.3	95.8±7.7	61.9±48.5	68.4±33.0
peg-ins	81.2±26.8	78.1±36.4	100.6±7.4	96.6±10.6	**100.4±76.9**	91.6±57.8	98.5±133.3	93.3±9.8	73.6±50.0	66.5±33.0
pick-pl	103.0±34.7	84.2±42.5	98.0±7.5	95.9±9.2	89.0±60.3	**116.9±130.1**	125.5±187.4	97.3±6.5	91.3±45.4	49.1±58.0
push	75.5±24.1	96.9±33.2	100.0±7.4	87.4±18.9	94.8±74.8	120.7±91.0	**119.4±178.1**	87.0±16.5	99.0±34.7	75.1±49.7
reach	76.4±29.7	87.0±40.3	99.7±8.6	99.8±4.1	101.7±82.9	107.6±97.3	184.3±302.8	**91.0±5.6**	76.6±45.9	64.6±56.3
win-close	83.9±26.2	92.7±30.5	99.2±7.9	98.7±4.8	104.2±85.2	97.1±68.2	109.2±159.3	94.0±8.7	**86.6±48.5**	87.3±39.8
win-open	86.3±35.5	64.5±41.3	99.9±7.4	87.0±16.5	97.4±81.8	87.6±60.2	152.9±228.2	94.5±8.0	59.6±39.6	**60.9±33.0**

Table 30: Retained return (% of baseline) after ablating the top 10% of neurons for Meta-World MT10 (400 units).

Ablated / Eval	btn-press	door-open	drawer-cl	drawer-op	peg-ins	pick-pl	push	reach	win-close	win-open
btn-press	**49.6±40.3**	42.0±45.0	99.6±7.8	74.7±26.4	109.8±102.9	108.9±98.1	115.2±176.1	76.4±18.2	79.2±47.1	82.3±21.9
door-open	65.9±37.9	**56.5±33.2**	100.9±7.9	88.7±9.1	105.1±84.3	95.6±90.9	62.6±79.0	83.9±13.5	73.8±47.4	91.1±20.5
drawer-cl	79.1±24.9	69.7±57.4	**100.0±7.0**	80.4±23.2	93.8±83.0	99.7±78.0	375.7±675.6	91.9±8.7	47.0±47.6	12.8±5.0
drawer-op	83.4±26.8	51.8±40.3	102.2±7.0	**79.0±13.5**	99.9±93.5	114.1±101.0	148.7±233.8	89.5±11.2	57.0±50.3	57.6±34.8
peg-ins	73.8±31.0	75.4±37.5	97.9±8.6	86.5±12.1	**93.7±67.8**	138.1±117.7	149.0±271.1	84.2±23.4	63.1±49.8	64.3±43.6
pick-pl	90.5±39.5	76.4±40.6	97.5±7.2	94.8±11.7	77.7±50.4	**137.2±109.7**	237.5±459.9	95.7±5.5	60.4±52.8	34.0±53.3
push	66.9±25.2	68.6±41.8	100.9±7.9	76.3±21.6	80.9±55.3	118.2±83.9	**114.6±161.3**	79.5±20.4	80.7±32.4	96.9±23.6
reach	72.5±31.1	76.2±49.6	100.5±7.7	91.4±9.8	108.1±85.1	114.8±80.9	150.7±218.0	**80.3±9.4**	64.1±42.0	44.5±39.5
win-close	83.4±29.5	81.5±37.0	98.2±7.7	94.1±13.1	112.6±93.1	90.7±60.8	70.5±95.0	85.5±13.5	**71.4±49.9**	89.0±42.2
win-open	83.6±39.5	33.2±17.4	100.3±7.5	78.8±18.4	113.5±99.6	91.9±67.6	159.4±298.9	88.7±10.8	60.9±37.3	**51.5±13.2**

Table 31: Retained return (% of baseline) after ablating the top 20% of neurons for Meta-World MT10 (400 units).

Ablated / Eval	btn-press	door-open	drawer-cl	drawer-op	peg-ins	pick-pl	push	reach	win-close	win-open
btn-press	**44.3±39.6**	50.7±40.7	99.5±7.7	60.0±12.6	111.2±105.0	104.7±99.1	101.0±142.5	66.9±16.0	65.0±52.0	73.4±26.9
door-open	58.6±35.0	**47.6±39.7**	97.1±7.2	69.2±29.1	97.2±83.1	100.5±101.5	64.0±80.2	71.5±23.8	83.5±36.2	73.6±37.5
drawer-cl	56.1±25.2	40.5±48.5	**69.9±37.6**	74.1±33.8	71.7±48.5	126.8±132.9	101.2±132.0	58.8±21.1	34.5±42.4	29.3±48.0
drawer-op	78.1±24.3	45.8±44.1	79.7±45.1	**59.0±30.5**	103.0±96.2	93.5±78.3	128.9±192.8	83.4±15.0	51.3±48.5	23.8±14.0
peg-ins	54.6±22.7	23.2±16.7	89.3±22.6	72.7±26.7	**113.1±84.9**	110.3±70.0	78.6±107.8	73.9±23.1	51.2±42.4	56.3±48.3
pick-pl	63.1±40.8	68.8±42.4	97.8±6.9	81.3±16.5	79.4±49.8	**133.3±88.5**	118.3±192.2	84.8±15.3	44.6±41.2	28.8±44.8
push	67.0±19.4	44.3±45.7	97.5±9.2	68.9±26.2	98.3±93.5	116.6±77.5	**87.0±104.0**	78.5±13.9	48.3±35.9	64.8±53.8
reach	65.2±30.7	70.1±42.3	101.5±7.1	79.7±25.3	90.4±59.6	96.7±68.1	83.0±109.5	**78.5±13.1**	50.2±36.1	43.1±39.5
win-close	60.4±41.7	70.7±40.8	96.7±7.3	91.2±19.1	114.9±95.3	96.3±68.3	86.3±124.5	85.8±13.6	**76.8±46.2**	79.7±26.2
win-open	62.1±23.6	37.4±37.2	103.3±6.1	75.8±20.7	112.1±103.8	81.5±58.0	105.0±144.7	85.7±13.8	51.8±39.7	**44.0±32.3**

Table 32: Retained return (% of baseline) after ablating the top 30% of neurons for Meta-World MT10 (400 units).

Ablated / Eval	btn-press	door-open	drawer-cl	drawer-op	peg-ins	pick-pl	push	reach	win-close	win-open
btn-press	**40.3±28.6**	48.6±43.9	99.9±7.4	55.6±20.3	112.2±102.4	96.8±78.2	80.4±129.6	49.4±16.1	72.0±39.8	57.5±40.2
door-open	50.5±38.5	**41.9±39.2**	100.9±7.9	81.2±23.5	103.1±84.9	126.5±149.5	56.9±71.8	63.5±22.2	78.7±47.1	65.9±42.3
drawer-cl	59.1±27.8	51.6±60.5	**68.4±45.3**	69.5±38.7	76.4±54.1	84.9±59.9	202.2±322.1	62.0±26.3	23.6±28.1	44.0±55.1
drawer-op	74.0±22.2	46.0±44.5	84.6±42.2	**50.0±38.6**	116.5±100.9	106.2±81.1	81.3±101.0	64.8±21.5	45.6±47.2	17.7±12.6
peg-ins	57.2±33.0	40.3±29.6	82.9±30.3	59.6±26.6	**110.1±84.8**	114.6±79.5	94.3±135.5	53.8±29.4	34.6±30.2	32.0±32.7
pick-pl	63.7±32.1	46.8±26.6	99.8±7.9	61.2±24.9	80.5±52.4	**127.1±88.7**	95.9±130.3	76.0±13.1	43.6±37.1	23.9±30.9
push	64.5±19.0	41.0±24.6	81.4±35.8	59.6±24.3	90.0±77.2	89.8±60.4	**205.3±351.8**	65.1±15.3	56.3±38.0	46.4±54.1
reach	53.7±25.1	41.5±42.7	100.7±7.1	70.5±32.3	98.4±87.9	109.7±90.2	151.2±275.0	**77.7±9.4**	37.1±29.4	48.4±42.6
win-close	53.5±40.2	61.2±54.5	97.3±9.0	79.4±27.4	124.5±104.9	92.7±65.9	84.3±129.2	67.2±17.4	**42.6±31.5**	77.0±39.5
win-open	59.5±23.9	46.9±38.6	102.0±7.3	82.7±20.5	96.4±76.5	83.2±59.7	147.9±267.2	69.5±24.2	55.2±49.8	**18.2±11.7**

Table 33: Retained return (% of baseline) after ablating the top 50% of neurons for Meta-World MT10 (400 units).

Ablated / Eval	btn-press	door-open	drawer-cl	drawer-op	peg-ins	pick-pl	push	reach	win-close	win-open
btn-press	**16.9±14.3**	40.3±46.2	98.4±8.2	55.1±30.6	112.5±100.7	99.5±78.0	58.8±73.8	34.2±17.8	26.6±34.8	44.1±45.6
door-open	43.6±27.9	**30.9±32.8**	87.2±34.1	31.2±11.3	80.0±49.8	104.4±73.0	60.9±84.8	50.5±20.4	39.0±39.8	55.0±41.2
drawer-cl	50.6±19.7	11.1±6.7	**58.4±45.4**	59.3±26.3	75.6±53.5	81.0±55.5	83.0±100.0	30.5±8.0	25.0±18.7	6.7±2.3
drawer-op	48.1±18.9	25.6±29.0	77.2±43.7	**53.6±25.8**	103.6±97.0	70.2±46.0	53.4±60.7	33.1±21.9	28.3±31.0	25.0±38.2
peg-ins	35.1±43.7	33.2±42.3	70.6±35.6	54.2±36.9	**138.2±122.3**	110.0±92.3	59.2±68.2	45.0±39.8	28.9±29.5	44.1±40.6
pick-pl	39.7±30.0	41.2±37.6	87.8±15.5	64.3±22.4	84.5±67.0	**92.3±73.7**	38.0±46.9	41.7±25.1	49.4±46.5	26.7±28.2
push	63.4±17.8	34.0±30.6	71.2±25.2	47.0±24.1	67.2±37.4	73.2±51.9	**91.1±118.2**	44.9±19.1	52.9±40.9	14.0±9.8
reach	43.0±22.8	59.6±41.1	98.7±9.2	45.7±17.9	116.1±105.1	80.0±60.7	138.6±208.8	**46.9±21.2**	51.9±50.0	34.6±38.9
win-close	54.3±21.6	26.9±38.4	92.7±6.5	60.0±19.5	128.4±97.2	70.0±43.0	55.3±89.2	51.1±18.4	**18.7±17.5**	39.4±46.8
win-open	40.0±32.9	29.7±38.2	100.0±7.4	64.4±24.4	77.1±52.1	88.1±81.9	48.8±59.1	50.6±26.5	25.4±22.7	**13.6±7.8**

E IMPORTANCE-GUIDED PRUNING

To test whether importance profiles are practically actionable, we compare importance-guided pruning against a matched random control on AntDir. For each task, we rank penultimate-layer neurons by that task's importance profile and zero the lowest-ranked fraction s. The control zeroes an equally sized, uniformly random set of neurons. Retention is the masked return on the same task as a percentage of the same seed's unmasked return, averaged over the eight directions. Table 34 reports mean $\pm$ std across the five seeds.

Table 34: Retention (% of unmasked return) in AntDir when zeroing the fraction s of penultimate-layer neurons with the lowest importance for the evaluated task (guided) or an equally sized random set (random), mean $\pm$ std across 5 seeds.

	64 units		400 units	
s	Guided	Random	Guided	Random
0.05	100 ± 1	78 ± 7	100 ± 1	100 ± 4
0.10	100 ± 0	65 ± 9	99 ± 1	91 ± 7
0.20	100 ± 1	45 ± 9	100 ± 1	84 ± 8
0.30	100 ± 1	31 ± 9	100 ± 2	71 ± 8
0.50	85 ± 6	23 ± 5	100 ± 1	47 ± 7

Guided pruning is lossless up to $s = 0.30$ at width 64 and up to $s = 0.50$ at width 400, where random pruning retains 31% and 47%. Guided retention is at least as high as random retention for every seed and sparsity level at width 64 (25 of 25) and for 21 of 25 at width 400. The four exceptions occur at $s \leq 0.10$, where both methods are at ceiling and differ by less than 3 points. One direction of one width-400 seed has an unmasked return of 422, far below every other direction (at least 1884), and is excluded because retention relative to a non-performing baseline is not meaningful.

Two boundaries apply. First, the masks are task-conditional. A guided mask preserves the task it was built for but damages the others, because neurons unimportant for one direction are important for another. At width 64 and $s = 0.30$, retention on the remaining seven directions falls to $49 \pm 7\%$. At width 400 and $s = 0.50$ it falls to $66 \pm 7\%$. Second, this is a proof of concept rather than a tuned pruning method. As the HalfCheetahVel results show, importance overlap does not guarantee functional separability, so pruning or routing built on importance profiles still requires a causal validation step such as the ablations used here.

F LAYER-WISE IMPORTANCE OVERLAP

The main analysis uses the penultimate layer, which feeds the linear action head directly. Its geometric organization could therefore be imposed by the readout rather than formed inside the network. To test this, we repeat the importance and BC analysis on the first hidden layer, with a random-initialization control computed separately for each layer. Table 35 reports the AntDir results for both encodings and both widths.

The angular decay is present at the first hidden layer in all four conditions and is monotonic in every seed. Under the continuous encoding the untrained control also decays, because similar directions produce correlated inputs. This input prior is stronger at the first layer than at the penultimate layer, yet training roughly doubles it at the first layer (0.237 vs. 0.111 at width 64, 0.300 vs. 0.128 at width 400) and quadruples it at the penultimate layer. Under the one-hot encoding the control is flat at both layers, yet the trained networks show a first-layer decay of 0.146 and 0.209. This is the layer where the one-hot input enters, so the organization is neither inherited from the input nor created by the readout. It is present at both hidden layers, and the gap to the untrained control is larger at the penultimate one.

In HalfCheetahVel the picture is weaker, as shown in Table 36. Velocity-distance decay, BC at distance 0.25 minus BC at 1.75, is present at both layers, but at the first hidden layer it does not clearly exceed the untrained control: 0.142 ± 0.030 against 0.117 ± 0.009, with three of five trained seeds inside the range of the random seeds. At the penultimate layer the trained decay is $0.253 \pm$

Table 35: AntDir mean BC by angular distance at the first hidden layer and the penultimate layer. Decay is BC at 45° minus BC at 180°, mean ± std across 5 seeds, for trained policies and for untrained networks at the same layer. Continuous-encoding values are recomputed on a fresh set of rollouts from the same policies and differ from Table 1 by at most 0.026 per bin. Monotone counts the trained seeds in which mean BC decreases strictly from 45° to 180°.

	Layer	45°	90°	135°	180°	Decay	Random decay	Monotone
64, continuous	First	0.934	0.824	0.731	0.697	0.237 ± 0.018	0.111 ± 0.008	5/5
	Penult.	0.884	0.729	0.619	0.578	0.305 ± 0.027	0.075 ± 0.016	5/5
400, continuous	First	0.910	0.765	0.651	0.610	0.300 ± 0.021	0.128 ± 0.008	5/5
	Penult.	0.828	0.614	0.481	0.440	0.387 ± 0.014	0.095 ± 0.007	5/5
64, one-hot	First	0.939	0.869	0.815	0.794	0.146 ± 0.026	-0.002 ± 0.004	5/5
	Penult.	0.897	0.798	0.733	0.702	0.195 ± 0.071	0.003 ± 0.005	5/5
400, one-hot	First	0.882	0.777	0.700	0.674	0.209 ± 0.021	0.000 ± 0.001	5/5
	Penult.	0.790	0.623	0.524	0.496	0.295 ± 0.020	0.000 ± 0.001	5/5

0.069 against 0.117 ± 0.010. The target velocity is a single scalar input, so the untrained prior is already substantial at the first layer and training adds little to it there. For HalfCheetahVel, the layer-wise result therefore holds only at the penultimate layer.

Table 36: HalfCheetahVel mean BC by velocity distance at the first hidden layer and the penultimate layer, for trained and untrained networks at the same layer. Decay is BC at distance 0.25 minus BC at 1.75, mean ± std across 5 seeds.

Layer		0.25	0.50	0.75	1.00	1.25	1.50	1.75	Decay
First	Trained	0.991	0.976	0.959	0.938	0.914	0.885	0.848	0.142 ± 0.030
	Random	0.992	0.983	0.973	0.960	0.943	0.918	0.876	0.117 ± 0.009
Penult.	Trained	0.975	0.941	0.907	0.867	0.822	0.773	0.722	0.253 ± 0.069
	Random	0.992	0.982	0.973	0.960	0.944	0.918	0.875	0.117 ± 0.010

G Sign Structure of Penultimate-Layer Importance

The absolute value in Section 3.1 prevents a neuron's positive and negative effects on different action dimensions from cancelling. A natural concern is that it might also hide sign structure, such as a neuron that pushes the ant toward one goal direction and away from the opposite one. At the penultimate layer the absolute value does not lose this information. The action head is linear, so $\partial a_d / \partial z_h = W[d, h]$ is the same for every observation and task, and ReLU activations satisfy $z_t[h] \geq 0$. Each signed contribution $I_t[d, h] = W[d, h]\, z_t[h]$ therefore keeps the sign of $W[d, h]$ at every timestep and under every task, and its magnitude is $|W[d, h]|\, z_t[h]$. The signed and absolute profiles therefore vary across tasks only through the activations. Each penultimate neuron has a fixed direction of effect on every action dimension, and tasks differ in how strongly they use it. Opponent or axis-like coding, in which the same neuron drives the action in opposite directions for opposite goals, would require a task-dependent Jacobian. In our architecture this first becomes possible at the first hidden layer, where downstream ReLU gating makes the Jacobian state-dependent. A signed analysis at that layer is left to future work.

H Meta-World MT10 Results

This appendix reports the analysis of the original MT10 policies, which motivated the seven-task subset in Section 4.3. These policies fail to learn three of the ten tasks: peg-insert-side, pick-place, and push remain at or below their random baselines, as shown in Table 37. Importance profiles for unlearned tasks do not reflect task-solving computation. The absence of stable structure in MT10 could therefore reflect either the lack of a continuous task parameterization or incomplete learning, so MT10 is at best a weak negative control.

In Figure 4 we can see the BC matrix for MT10. The values are generally lower than in previous environments. Some pairs of tasks do show greater overlap than others, such as drawer-close

Table 37: Mean episodic return per task in Meta-World MT10 for the trained policy and a random baseline (400 hidden units, 5 seeds).

Task	Trained (mean±std)	Random (mean±std)
button-press-topdown	1988.2 ± 485.7	5.3 ± 10.2
door-open	3618.4 ± 997.5	279.4 ± 18.0
drawer-close	4637.1 ± 270.6	162.9 ± 356.7
drawer-open	2531.3 ± 77.2	618.6 ± 15.7
peg-insert-side	4.6 ± 2.6	8.9 ± 0.3
pick-place	4.2 ± 2.3	8.2 ± 1.1
push	31.8 ± 32.9	26.6 ± 1.1
reach	4634.9 ± 199.6	724.1 ± 17.4
window-close	3452.3 ± 969.4	256.5 ± 8.9
window-open	3091.1 ± 601.2	205.3 ± 0.9

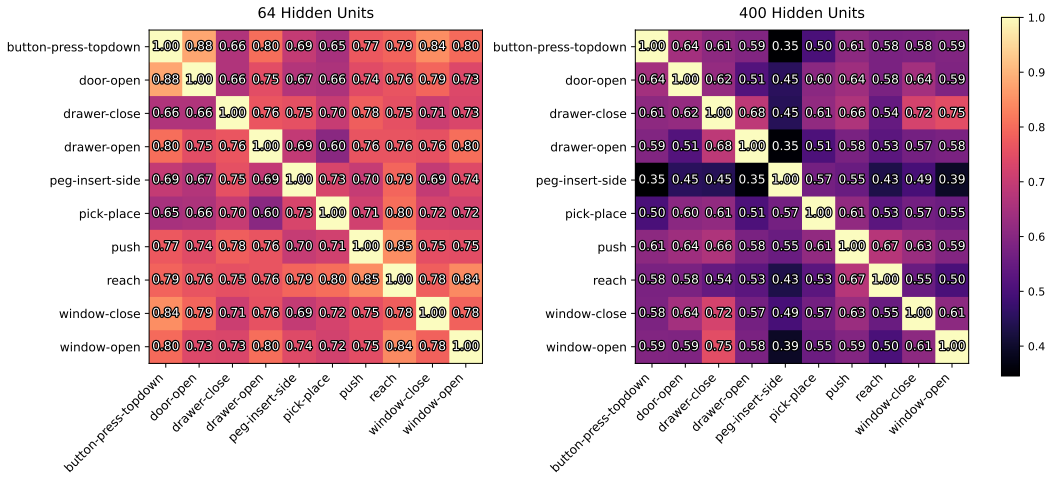

Figure 4: Mean Bhattacharyya Coefficient (BC) between neuron importance profiles for different task across all five seeds in the MetaWorld/MT10 environment. On the left, the matrix for the experiment with 64 hidden units per layer; on the right, the matrix for 400 hidden units.

and window-open for the 400 unit variant, or door-open and button-press for the 64 unit variant. However, there is no consistency between the different architectures. Furthermore, if we ignore the environments where the policy did not learn, it is still difficult to find any pattern.

Considering that ablating or evaluating ablation of task profiles that do not actively solve the task is not very informative, we focus on the tasks that were solved. In Table 38 we can see the ablation results for the 400 hidden unit variant on tasks that surpass the baseline beyond their standard deviations.

I Meta-World MT10 Training Duration

We trained Meta-World MT10 at two durations: 20 million and 40 million timesteps. Table 39 shows per-task mean return across 5 seeds. Doubling the training budget did not improve performance: 9 of 10 tasks show equal or lower return at 40M steps, with door-open, window-close, and window-open suffering the largest drops. This is consistent with catastrophic interference in multi-task settings. We therefore report results at 20 million timesteps, which matches 2 million steps per task as used by McLean et al. (2025a) for single-task benchmarking.

Table 38: Retained return as % of baseline when ablating the top 5% of neurons ranked by importance for each source task (rows) and evaluating on all tasks (columns) in Meta-World MT10. Diagonal entries (bold) reflect self-damage.

Ablated /Eval	button-press	door-open	drawer-close	drawer-open	reach	window-close	window-open
button-press	**80.4±35.9**	56.5±41.9	99.5±7.8	88.2±16.5	93.8±6.6	80.8±37.9	83.4±23.7
door-open	63.0±34.5	**67.5±54.0**	100.4±8.5	92.0±11.5	93.8±8.3	63.5±47.0	83.9±24.3
drawer-close	84.9±26.9	76.8±51.5	**98.9±7.0**	81.4±27.4	94.8±7.8	51.3±54.1	31.5±30.8
drawer-open	84.7±31.0	54.5±31.4	101.2±7.2	**81.6±24.3**	95.8±7.7	61.9±48.5	68.4±33.0
reach	76.4±29.7	87.0±40.3	99.7±8.6	99.8±4.1	**91.1±5.6**	76.6±45.9	64.6±56.3
window-close	83.9±26.2	92.7±30.5	99.2±7.9	98.7±4.8	94.0±8.7	**86.6±48.5**	87.3±39.8
window-open	86.3±35.5	64.5±41.3	99.9±7.4	87.0±16.5	94.5±8.0	59.6±39.6	**60.9±33.0**

Table 39: Meta-World MT10 mean return (± std across 5 seeds) at 20M and 40M training timesteps (64-unit network). Bold indicates the higher value for each task.

Task	20M steps	40M steps	Random
button-press-topdown	$\mathbf{1901.8 \pm 608.9}$	1297.5 ± 179.1	21.4 ± 20.1
door-open	$\mathbf{3062.5 \pm 901.8}$	969.3 ± 1017.1	281.3 ± 3.3
drawer-close	$\mathbf{4768.8 \pm 88.8}$	4717.6 ± 147.7	6.2 ± 8.1
drawer-open	$\mathbf{2496.7 \pm 49.5}$	2243.5 ± 274.1	615.7 ± 10.3
peg-insert-side	$\mathbf{24.8 \pm 27.1}$	6.4 ± 3.1	8.9 ± 0.1
pick-place	6.0 ± 2.4	$\mathbf{6.1 \pm 2.8}$	8.3 ± 1.3
push	$\mathbf{155.7 \pm 172.5}$	48.0 ± 49.6	27.5 ± 1.9
reach	$\mathbf{3592.9 \pm 1070.7}$	3140.1 ± 630.3	729.6 ± 29.3
window-close	$\mathbf{2755.3 \pm 1507.1}$	1296.9 ± 1138.9	257.5 ± 3.3
window-open	$\mathbf{3362.0 \pm 850.9}$	1274.7 ± 1220.0	205.6 ± 0.8

J Compute Resources

All experiments were run on a single GPU instance of an NVIDIA H100. A single training run (one seed, one environment, one architecture width) takes approximately two hours. The GPU can accommodate multiple seeds in parallel. With 3 environments $\times$ 2 widths $\times$ 5 seeds $=$ 30 runs, total sequential compute is roughly 60 GPU-hours; actual wall-clock time is lower due to parallelism. The analysis pipeline (importance profiling, BC computation, and ablation evaluation) adds another 40 minutes per run, however, it does not require GPU and is only capped by the CPU count. We used 32 CPU cores on all jobs.