Task-specific Subnetwork Discovery in Reinforcement Learning for Autonomous Underwater Navigation

Yi-Ling Liu*, Melvin Laux*†, Mariela De Lucas Alvarez*, Frank Kirchner*†, Rebecca Adam*
* Robotics Innovation Center, German Research Center for Artificial Intelligence, Bremen, Germany
† Robotics Research Group, University of Bremen, Bremen, Germany
Correspondence: yi-ling.liu@dfki.de

***Abstract*—Autonomous underwater vehicles are required to perform multiple tasks adaptively and in an explainable manner under dynamic, uncertain conditions and limited sensing, challenges that classical controllers struggle to address. This demands robust, generalizable, and inherently interpretable control policies for reliable long-term monitoring. Reinforcement learning, particularly multi-task RL, overcomes these limitations by leveraging shared representations to enable efficient adaptation across tasks and environments. However, while such policies show promising results in simulation and controlled experiments, they yet remain opaque and offer limited insight into the agent's internal decision-making, creating gaps in transparency, trust, and safety that hinder real-world deployment. The internal policy structure and task-specific specialization remain poorly understood. To address these gaps, we analyze the internal structure of a pretrained multi-task reinforcement learning network in the HoloOcean simulator for underwater navigation by identifying and comparing task-specific subnetworks responsible for navigating toward different species. We find that in a contextual multi-task reinforcement learning setting with related tasks, the network uses only about 1.5% of its weights to differentiate between tasks. Of these, approximately 85% connect the context-variable nodes in the input layer to the next hidden layer, highlighting the importance of context variables in such settings. Our approach provides insights into shared and specialized network components, useful for efficient model editing, transfer learning, and continual learning for underwater monitoring through a contextual multi-task reinforcement learning method.**

***Index Terms*—AUVs, explainable reinforcement learning, mechanistic interpretability, multi-task reinforcement learning, pruning**

I. Introduction

Reinforcement learning (RL) offers the potential for autonomous underwater vehicles to adaptively navigate complex environments, where traditional model-based control methods struggle due to disturbances, partial observability and changing environmental conditions [7].

Although RL for robotic navigation has advanced recently, trust and safe deployment in real-world missions are limited by the lack of interpretability and incomplete understanding of internal decision-making processes, which is particularly critical for autonomous underwater vehicle (AUV) control models. For long-term underwater monitoring tasks, where failures could result in mission loss or environmental risk, addressing this challenge is essential.

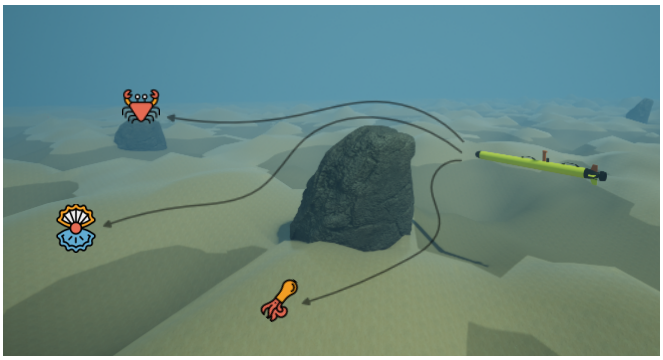


Fig. 1: The navigation task is simulated in HoloOcean. For the specified species in each task, the AUV should navigate to find the crab, the shell or the octopus.

Concurrently, multi-task reinforcement learning (MTRL) is theoretically expected to exploit shared knowledge across related tasks [10], thereby enhancing generalization, improving data efficiency, and increasing robustness to variable conditions, which are highly advantageous in the uncertain and non-stationary dynamics of underwater environments. Contextual MTRL further allows task specification through context variables or the incorporation of environment-specific information, such as currents, which are known to influence the underlying Markov decision processes (MDPs).

A. Related Work

Despite theoretical and empirical evidence suggesting that MTRL benefits from shared knowledge across tasks [10], there has been limited direct investigation into the internal structure of these networks. This gap is particularly notable from a mechanistic interpretability [3] perspective. Even in the single-task reinforcement learning setting, explainability research remains relatively underexplored compared to other areas of machine learning [20].

With the recent rise of mechanistic interpretability, driven by advances in large language models, there has been growing interest in analyzing the internal mechanisms of reinforcement learning systems. However, existing work remains limited to specialized settings and architectures. Prior studies primarily investigate toy problems, such as goal misgeneralization in

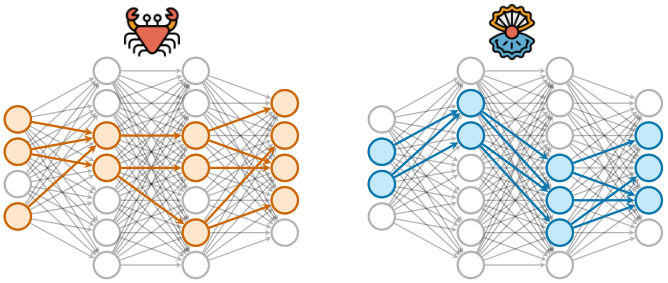


Fig. 2: Task-specific subnetworks specialized for navigation to the crab and to the shell, respectively.

maze-solving tasks [24], planning in simplified box-pushing environments [4], and memory usage in RNN-based world models in the MiniGrid Switching Memory environment [23]. To the best of our knowledge, no prior work has examined the internal structure or mechanisms of contextual MTRL networks.

B. Research Gap

Motivated by these considerations, we investigate a pretrained MTRL network presented in [17] to examine whether its parameters are effectively shared across tasks. In this paper, we further aim to identify task-specific subnetworks within the trained MTRL network that are sufficient for individual navigation tasks, such as reaching distinct target locations or monitor specific marine species. Characterizing these subnetworks can reveal failure modes, enable more efficient inference, and facilitate knowledge transfer across tasks. Ultimately, this contributes to the development of safer, more reliable, and computationally efficient autonomous navigation policies for underwater applications.

In summary, we address following research gaps:

- AUV control model lacks interpretability, which is needed for safe real-world deployment.
- It remains an open question whether contextual MTRL encodes common knowledge across related tasks or acquires it independently for each task.
- It is unclear whether contextual MTRL relies on context variables or other parameters to differentiate tasks.

C. Research Questions

To address these gaps, We seek to answer the following research questions through this study:

- RQ1: How are parameters shared in contextual MTRL for AUV control?
- RQ2: How are task-specific context variables associated with their respective tasks in contextual MTRL?

To address these research questions, we employ a pretrained Double DQN [15] value network for underwater navigation in [17]. For each task, the network is pruned to obtain a task-specific subnetwork, and the overlap between these subnetworks is analyzed to understand shared and task-specific components across different navigation tasks. Building on prior work in neural network modularity and subnetwork discovery, we hypothesize that shared components encode critical knowledge for generalization, while specialized components encode specific strategies. Initial experiments were conducted in MiniGrid [5] to establish the methodological feasibility and then extended to underwater navigation tasks using the HoloOcean simulator [22]. By comparing subnetworks across tasks, we identify task-specific structures that differentiate tasks in MTRL, while also uncovering important shared structures that can guide future transfer learning and the reuse of learned knowledge.

D. Novel Contribution

In summary, the main contributions of this work are:

- Demonstrating that MTRL utilizes a large portion of network weights for shared knowledge across tasks, indicating effective knowledge sharing as intended.
- Identifying that only a relatively small portion of weights are task-specific, highlighting the minimal task-dependent specialization required for individual objectives.
- Revealing the importance of context variables in MTRL, which enable the network to differentiate between related tasks effectively.

E. Notational Conventions

In this work we use bold lower case letters to denote vectors. Furthermore, $\odot$ denotes the element-wise multiplication (Hadamard Product), and we denote binary sets by $\mathbb{B} = \{0, 1\}$. Finally $\mathbb{1}_{\text{condition}}$ denotes the indicator function to represent binary decisions

$$\mathbb{1}_{\text{condition}} = \begin{cases} 1, & \text{if the condition is true,} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

It converts a logical condition into a binary value which we use to denote whether a given element satisfies a specific criterion.

II. Related Work

Recent research has shown that large neural networks often contain smaller functional subnetworks capable of performing specific tasks, suggesting that complex models may internally organize into modular computational structures.

A. Lottery Ticket Hypothesis

The Lottery Ticket Hypothesis [12] demonstrated that dense neural networks contain sparse subnetworks that can be trained in isolation from their initial weights while achieving comparable performance to the original model. Subsequent work provided theoretical insights into this phenomenon, showing that effective subnetworks can emerge through pruning processes and may explain why overparameterized networks generalize well [19].

Later studies further suggested that multiple performant subnetworks may coexist within a single model, supporting the idea that networks may encode different capabilities in partially independent structures [11].

B. Emergent Modularity

Complementary studies attempt to identify modules responsible for specific subtasks in trained full networks through weight pruning. Csordás et al. (2021) [9] proposed learning differentiable binary masks over weights using a Gumbel-Sigmoid relaxation [16], allowing the discovery of functional modules within trained networks. However, most subnetworks were not reused across tasks, indicating a lack of compositionality. Later work [18] showed that language and vision models often naturally exhibit structural compositionality without explicit guidance, decomposing tasks into modular subnetworks, suggesting that neural networks can learn compositionality.

These studies provide mixed evidence regarding functional modularity in neural networks, motivating further investigation into whether task-specific subnetworks also arise within MTRL networks.

C. Circuit Discovery

A closely related line of research is circuit discovery within the emerging field of mechanistic interpretability [3]. Circuits refer to sparse computational subgraphs of a model that encode specific aspects of its behavior. Circuit discovery aims to find which parts of a neural network are causally responsible for a specific behavior. Most existing work focuses on pretrained large language models [8].

However, the proposed methods are primarily tailored to transformer architectures, typically operating by pruning connections between high-level components such as entire MLP blocks or attention heads. As a result, they are not directly applicable to simpler or fundamentally different network architectures.

In RL, this research direction remains relatively underexplored. Sobotka et al. (2025) [23] investigate the internal representations of recurrent RL agents, identifying the minimal subnetwork responsible for memory-based behavior in DreamerV3 [13], a model-based RL agent with a GRU-based recurrent backbone [6].

Nevertheless, research on the internal mechanisms of RL agents remains limited, with most existing work focusing on specific architectures. Canonical model-free algorithms (e.g., DQN, PPO, SAC) remain poorly understood, and further work is needed to clarify their internal computation and decision-making. In addition, MTRL, despite its potential for shared representation learning, is still underexplored from the perspective of interpretability and needs further investigation.

III. PRELIMINARIES

A. Contextual Markov Decision Process and Multi-Task Reinforcement Learning

A contextual MDP [14, 21] is defined as a tuple $(\mathcal{C}, \mathcal{S}, \mathcal{A}, \mathcal{M})$, where $\mathcal{C}$ denotes a context space, $\mathcal{S}$ and $\mathcal{A}$ are shared state and action spaces, and $\mathcal{M}$ is a mapping from contexts to tasks. For each context $c \in \mathcal{C}$, this mapping specifies a task $\mathcal{T}_c = (\mathcal{S}, \mathcal{A}, P_c, R_c)$, where the transition dynamics P_c and reward function R_c may vary across contexts.

In MTRL, the objective is to learn a single model that performs well across a distribution of such tasks [25]. This is commonly achieved by conditioning the policy or value function on the context, e.g., learning a Q-function $Q(s, a, c)$ that captures task-dependent behavior. Training proceeds by sampling trajectories from multiple contexts and optimizing a joint objective over the task distribution, enabling the agent to share knowledge and generalize across related tasks.

IV. METHODOLOGY

We formulate underwater navigation as a contextual MTRL problem as in [17]. The tasks are encoded as a one-hot vector, with each variable representing a navigation goal toward a specific species. Unlike traditional RL, which learns each task independently, MTRL trains a single agent to solve multiple tasks simultaneously by leveraging shared knowledge. In theory, this enables more efficient learning and better generalization through shared representations. To answer our research questions in Section I-C, we identify task-associated subnetworks and analyze their shared and task-specific structures.

A. Identifying Task-specific Subnetworks

We identify task-specific subnetworks in a pretrained Q-network at the level of individual weights. To isolate subnetworks defined by the network parameters, we follow subnetwork discovery approaches similar to [23], [9], and [1], where binary masks are learned over pretrained, frozen weights. For the ease of subsequent notation, we conveniently flatten the complete Q-network's concatenated weight matrices including weights across all layers to a single vector formulation $\mathbf{w} = [w_1, \ldots, w_N] \in \mathbb{R}^{N \times 1}$, where N is the total number of network weights. Furthermore, we define a corresponding binary mask vector $\mathbf{m} = [m_1, \ldots, m_N] \in \mathbb{B}^{N \times 1}$ that is multiplied element-wise to the weights to produce the final masked parameters $\mathbf{w}' \in \mathbb{R}^{N \times 1}$:

$$\mathbf{w}' = \mathbf{w} \odot \mathbf{m}. \tag{2}$$

Here, each mask element $m_i \in \{1, \ldots, N\}$ in (2) is given by

$$m_i = \mathbb{1}_{p_i > 0.5} = \begin{cases} 1, & p_i > 0.5, \\ 0, & \text{else.} \end{cases} \tag{3}$$

The corresponding masking probability p_i in (3) depends on learnable parameters logit $l_i \in \mathbb{R}$ and is determined by the sigmoid function such that:

$$p_i = \sigma(l_i) = \frac{1}{1 + e^{-l_i}}. \tag{4}$$

The mask logits $\mathbf{l}$ are initialized with small Gaussian noise and constitute the only trainable parameters. Note that, during training the formulation $p_i = \sigma(l_i)$ and l_i serve the purpose of providing a differentiable parametrization for gradient-based optimization in the back-propagation step. Importantly, masks are learned exclusively for the weights, while all bias parameters are kept fixed and remain unmasked.

The mask parameters are optimized by sampling states directly from the RL state space with the corresponding task context. This enables the extraction of a sparse subnetwork that preserves the Q-value estimation of the original network on the target task, yielding a compact set of task-relevant connections, as illustrated in Fig. 2.

B. Mask Training Pipeline

To enable gradient-based optimization of binary masks, differentiable mask training relies on a continuous relaxation of the discrete masking operation. This is necessary because binary variables are non-differentiable and cannot be directly optimized with gradient descent. The method therefore optimizes continuous mask parameters and bridges the gap between discrete forward computation and continuous optimization using the straight-through estimator [2]. This applies hard binary decisions in the forward pass while using a differentiable surrogate during backpropagation, enabling learning of discrete masks within standard gradient-based training.

a) Mask Training: The sigmoid output $\sigma(l_i)$ serves as a differentiable relaxation of the binary mask, allowing gradients to flow through the masking process. As a result, optimization is performed over the unconstrained logits l_i rather than probabilities, which ensures stable gradient-based updates while maintaining $p_i \in [0, 1]$.

During training, masked weights are computed using the straight-through estimator:

$$\tilde{w}_i = w_i \Big([m_i - \sigma(l_i)]_{\text{stop}} + \sigma(l_i) \Big). \tag{5}$$

Here, w_i denotes pretrained network weights, which remain fixed throughout optimization. The stop-gradient operator indicates that the term inside $[\cdot]_{\text{stop}}$ is used only in the forward pass and does not contribute to gradients during backpropagation. This separates the discrete masking decision from the gradient-based optimization path, which instead flows through the differentiable surrogate $\sigma(l_i)$. As a result, only the logits l_i are updated, and the binary mask is learned indirectly via its continuous parameterization.

b) Objective: The mask parameters are learned by employing a combined loss L_{final} balancing task performance and sparsity:

$$L_{\text{final}} = L_{\text{q-value}} + \lambda L_{\text{sparsity}}, \tag{6}$$

where λ controls the sparsity penalty.

The Q-value loss enforces consistency between the original and masked networks:

$$L_{\text{q-value}} = \mathbb{E}\Big[\big(Q(s,a) - Q_{\text{masked}}(s,a) \big)^2 \Big]. \tag{7}$$

Sparsity is encouraged via an ℓ_1 penalty on the mask probabilities:

$$L_{\text{sparsity}} = \sum_{i=1}^{N} \sigma(l_i). \tag{8}$$

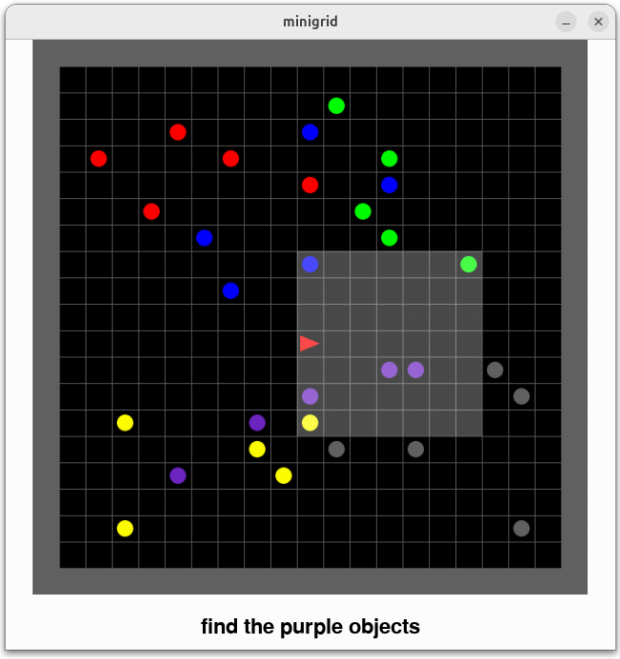


Fig. 3: In the Minigrid environment, the agent represented by a red triangle navigates to collect objects of a specified color. The shaded rectangle overlay denotes the agent's field of view.

c) Subnetwork Extraction: After training, the continuous mask is converted into a deterministic binary mask by thresholding the sigmoid of the learnable logit l_i with Equation (3).

Applying this mask with Equation (2) yields a sparse subnetwork, which retains task-relevant connections while removing redundant parameters by assigning them to zero.

V. Experiments

A. Task Definition

Two experimental environments were used to evaluate our method: a simplified Minigrid environment and an underwater navigation simulation in HoloOcean. We utilize a pretrained network in [17] and follow its experimental configuration, in which different marine species are represented as colored objects and encoded in the context variables using one-hot encoding.

1) Navigation in Minigrid: A toy navigation task in the Minigrid environment, as shown in Fig. 3, was employed to develop and test the method. In this task, the agent, represented as a red triangle, should navigate the environment to collect objects of the specified color. The shaded rectangle overlay in the figure indicates the agent's field of view, representing its partial observation of the environment. For subnetwork discovery experiments, we used a pretrained MTRL network trained on different tasks jointly for collecting red, blue, purple, and grey objects, with each color represented as a one-hot context variable.

2) Underwater Navigation in HoloOcean: In the underwater navigation tasks simulated in HoloOcean, as illustrated in

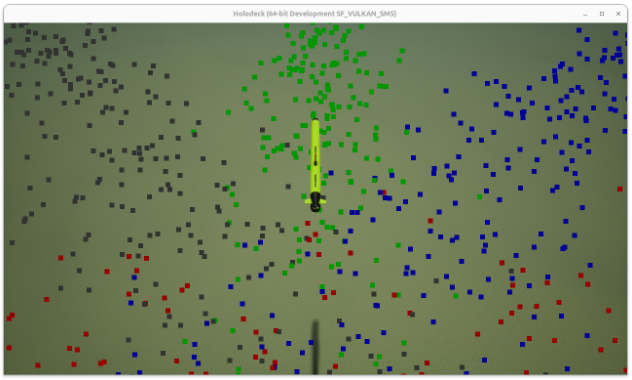


Fig. 4: In the HoloOcean environment, the yellow AUV navigates through the environment to pass by specified species, represented as squares of different colors, with each color corresponding to a distinct species.

Fig. 1, the agent should navigate the AUV from a starting position to find a specified species, such as crabs, shells, or octopuses. Only finding the species matching the context variable is considered correct. Experiments are conducted using various target species and spatial configurations to analyze how the network leverages shared knowledge across tasks. In our experimental environment, a variety of colored objects are used, as shown in Fig. 4, with each color representing a distinct species.

B. Experimental Analysis

Our experiments aim to understand the internal structural of networks in contextual MTRL, focusing on three main aspects: the performance of task-specific subnetworks, the distribution of shared weights across tasks, and the role of context variables. Analyses are conducted separately for the Minigrid and HoloOcean environments.

1) Subnetworks Performance: We identify subnetworks responsible for navigation toward different goals and set all weights not belonging to each selected subnetwork to zero. We then evaluate performance by comparing the normalized RL return of each subnetwork with that of the original full network on the task on which it was trained. The average return is computed as the total reward accumulated across all evaluation episodes, divided by the number of episodes. This quantity is subsequently normalized using the minimum and maximum achievable returns in each environment, respectively.

2) Shared Weights across Tasks: Subnetworks are compared across tasks to identify weights used by multiple tasks, revealing where shared knowledge in MTRL is represented. We visualize partially shared weights, which are activated in only some subnetworks, as well as globally shared weights, which are used across all subnetworks.

3) Context Variables: The usage of context variables within subnetworks is analyzed, along with subnetwork performance on other tasks, revealing the role of context in task identity.

C. Experimental Results

1) Subnetworks Performance: A comparison of normalized average returns between the full network and task-specific subnetworks on their corresponding pruning tasks indicates that the extracted subnetworks preserve task-relevant knowledge.

In MiniGrid, the subnetworks generally achieve their highest performance on the tasks for which they were derived, while exhibiting reduced performance on other tasks, consistent with effective task specialization. However, this trend does not hold uniformly across all subnetworks, as the blue subnetwork in particular shows weaker and less consistent performance. In contrast, the subnetworks corresponding to the purple and red tasks exhibit minimal performance degradation on their respective tasks, indicating more stable task-specific representations.

For the navigation task in HoloOcean, the full network exhibits a tendency toward a suboptimal policy with largely uniform performance across tasks. Nevertheless, the corresponding pruned subnetworks retain comparable performance, suggesting that pruning does not degrade the learned behavior and may instead isolate the most relevant task-specific structure.

Overall, performance on non-corresponding tasks decreases more noticeably than on the corresponding target task after applying the learned masks, in the case of a well-trained full network. Otherwise, when the full network is not well optimized, the pruned subnetworks also do not exhibit a marked performance degradation. The corresponding performance results are shown in Fig. 5.

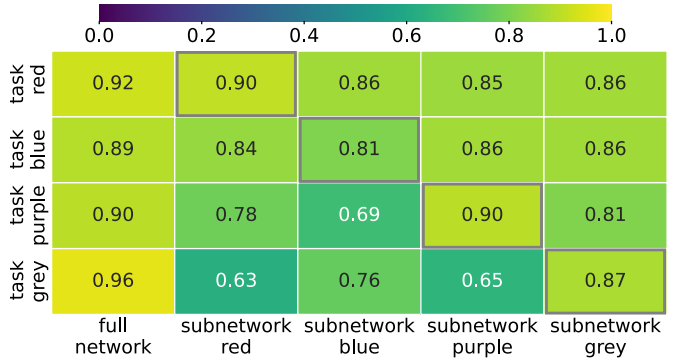
D. Shared Weights across Tasks

An analysis of shared and task-specific weights across subnetworks for the navigation task in the MiniGrid and HoloOcean environments is presented in Fig. 6 and Fig. 7, respectively. Following the pruning procedure, approximately 12.82% and 33.37% of the weights are removed from the navigation networks in MiniGrid and HoloOcean, respectively. After eliminating inactive weights across all tasks, globally shared parameters constitute approximately 96.84% and 98.23% of the total weights in the respective environments, whereas task-specific parameters account for only 1.58% and 1.45%.

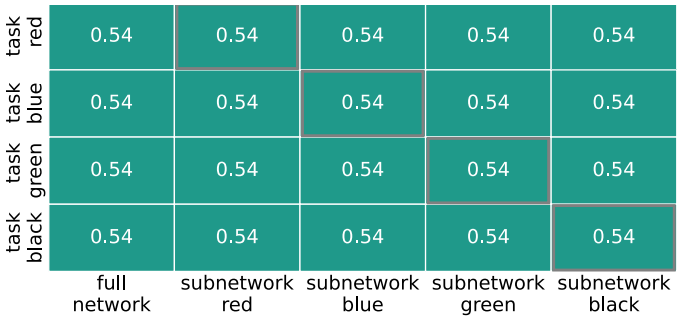
These results indicate that each task utilizes only a small subset of the overall network capacity, while the majority of parameters are devoted to shared representations. Moreover, this extensive parameter sharing suggests that the learned representations capture largely non-redundant common structure; otherwise, such weights would likely have been pruned. Overall, these findings indicate that the MTRL networks exploit substantial shared knowledge across closely related tasks.

E. Context Variables

Fig. 6 and Fig. 7 also show that each subnetwork predominantly depends on its associated context variable, underscoring



(a) MiniGrid normalized average return



(b) HoloOcean normalized average return

Fig. 5: A comparison of normalized average returns between the full network and task-specific subnetworks on their corresponding pruning tasks indicates that the extracted subnetworks preserve task-specific knowledge.

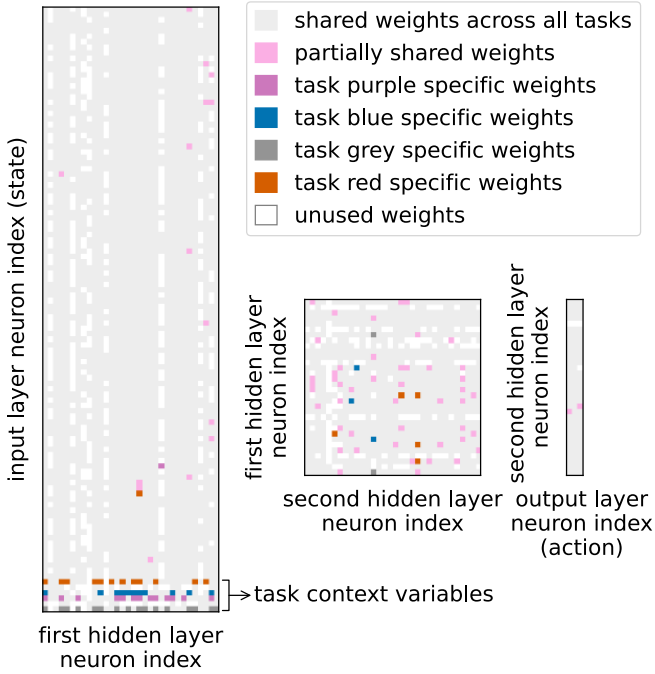


Fig. 6: The analysis of shared and task-specific weights across subnetworks of the Minigrid navigation task shows that most parameters are devoted to shared representations, while only a small fraction is task-specific. This suggests that the model learns largely non-redundant common structure across tasks. At the same time, each subnetwork strongly depends on its associated context variable, indicating that contextual information plays a key role in distinguishing tasks and guiding task-specific behavior within the MTRL framework.

the critical role of contextual information in shaping task-specific behavior. Within the task-specific parameters, approximately 81.25% and 85.46% correspond to weights connecting the context variables.

These results further indicate that connections from context variables of trained tasks to hidden-layer neurons are significantly more likely to be classified as task-specific than weights associated with other state inputs. Overall, this suggests that the pretrained network effectively leverages context variables to differentiate between tasks, consistent with the objectives of contextual MTRL.

VI. Conclusion

The experimental results demonstrate that the pruning process successfully identifies task-relevant subnetworks across most subtasks, with only a slight reduction in task-specific performance. Weight analysis further shows that these subnetworks strongly depend on their associated context variables, highlighting the crucial role of contextual information in guiding task-specific behavior and in distinguishing and exploiting task-relevant information within the contextual MTRL framework. Moreover, a substantial proportion of the retained parameters are shared across tasks, suggesting that the MTRL framework effectively leverages shared representations among related subtasks. This parameter sharing likely contributes to improved sample efficiency and enhanced generalization capability.

In addition, it is observed that several rows and columns of the weight matrices are substantially pruned, indicating that the corresponding neurons contribute minimally to task performance and are therefore less relevant. This finding motivates further investigation into neuron-level pruning and its relationship to task specialization.

Future work may extend this analysis by performing explicit neuron pruning and conducting an ablation study in which weights connecting context variables in the original full network are randomly removed to further validate their functional role. Such an investigation would enable a systematic assessment of the influence and importance of these connections on task-specific performance, thereby providing deeper insights into the role of context variables in shaping behavior within the contextual MTRL framework.

Acknowledgments

This work was funded by the German Federal Ministry for the Environment, Climate Action, Nature Conversation and Nuclear Safety (BMUKN) supported by the ZUG under grants 67KIA4036C and 67KIA4036A, and partially supported by

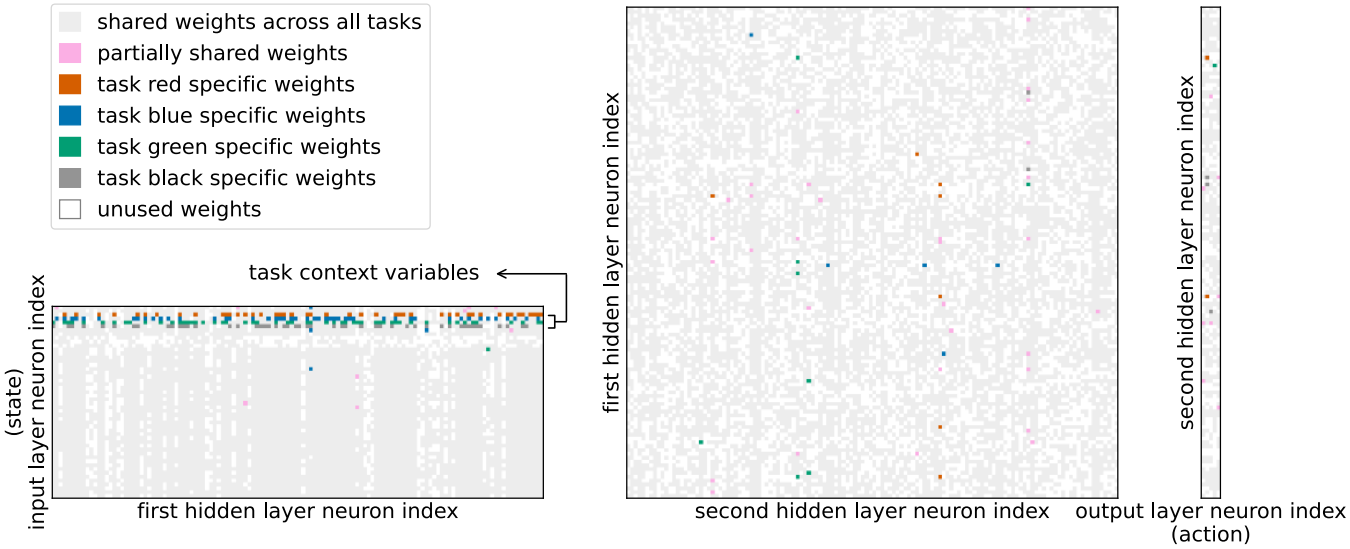


Fig. 7: The comparison of weights across subnetworks in the HoloOcean navigation task shows that, after pruning, most parameters are shared, with only a small fraction being task-specific. This suggests that the model learns a largely non-redundant shared representation across tasks, encoded in the shared weights. Moreover, each subnetwork strongly depends on its corresponding task context variable, highlighting the key role of contextual information in distinguishing tasks and guiding behavior within the MTRL framework.

the German Federal Ministry of Research, Technology and Space (BMFTR) under the Robotics Institute Germany (RIG) under grant 16ME1010. The authors would like to thank Alexander Fabisch, Yuhan Jin, and Nayari Lessa for their valuable feedback and discussion on this manuscript.

References

[1] Deniz Bayazit et al. "Discovering Knowledge-Critical Subnetworks in Pretrained Language Models". In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. EMNLP 2024. Ed. by Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen. Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 6549–6583. DOI: 10.18653/v1/2024.emnlp-main.376. URL: https://aclanthology.org/2024.emnlp-main.376/.

[2] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. *Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation*. Aug. 15, 2013. DOI: 10.48550/arXiv.1308.3432. arXiv: 1308.3432 [cs]. URL: http://arxiv.org/abs/1308.3432.

[3] Leonard Bereska and Stratis Gavves. "Mechanistic Interpretability for AI Safety - A Review". In: *Transactions on Machine Learning Research* (Apr. 27, 2024). ISSN: 2835-8856. URL: https://openreview.net/forum?id=ePUVetPKu6.

[4] Thomas Bush et al. "Interpreting Emergent Planning in Model-Free Reinforcement Learning". In: *The Thirteenth International Conference on Learning Representations*. 2025. URL: https://openreview.net/forum?id=DzGe40glxs.

[5] Maxime Chevalier-Boisvert et al. "Minigrid & Miniworld: Modular & Customizable Reinforcement Learning Environments for Goal-Oriented Tasks". In: *CoRR* abs/2306.13831 (2023).

[6] Kyunghyun Cho et al. "Learning Phrase Representations Using RNN Encoder–Decoder for Statistical Machine Translation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. EMNLP 2014. Ed. by Alessandro Moschitti, Bo Pang, and Walter Daelemans. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1724–1734. DOI: 10.3115/v1/D14-1179. URL: https://aclanthology.org/D14-1179/.

[7] Leif Christensen et al. "Recent Advances in AI for Navigation and Control of Underwater Robots". In: *Current Robotics Reports* 3.4 (Dec. 1, 2022), pp. 165–175. ISSN: 2662-4087. DOI: 10.1007/s43154-022-00088-3. URL: https://doi.org/10.1007/s43154-022-00088-3.

[8] Arthur Conmy et al. "Towards Automated Circuit Discovery for Mechanistic Interpretability". In: *Advances in Neural Information Processing Systems*. Ed. by A. Oh et al. Vol. 36. Curran Associates, Inc., 2023, pp. 16318–16352. URL: https://proceedings.neurips.cc/paper_files/paper/2023/file/34e1dbe95d34d7ebaf99b9bcaeb5b2be-Paper-Conference.pdf.

[9] Róbert Csordás, Sjoerd van Steenkiste, and Jürgen Schmidhuber. "Are Neural Nets Modular? Inspecting Functional Modularity Through Differentiable Weight Masks". In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=7uVcpu-gMD.

[10] Carlo D'Eramo et al. "Sharing Knowledge in Multi-Task Deep Reinforcement Learning". In: International Conference on Learning Representations. Sept. 23, 2019. URL: https://openreview.net/forum?id=rkgpv2VFvr.

[11] James Diffenderfer and Bhavya Kailkhura. "Multi-Prize Lottery Ticket Hypothesis: Finding Accurate Binary Neural Networks by Pruning A Randomly Weighted Network". In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=U_mat0b9iv.

[12] Jonathan Frankle and Michael Carbin. "The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks". In: *International Conference on Learning Representations*. 2019. URL: https://openreview.net/forum?id=rJl-b3RcF7.

[13] Danijar Hafner et al. *Mastering Diverse Domains through World Models*. Apr. 17, 2024. DOI: 10.48550/arXiv.2301.04104. arXiv: 2301.04104 [cs]. URL: http://arxiv.org/abs/2301.04104.

[14] Assaf Hallak, Dotan Di Castro, and Shie Mannor. *Contextual Markov Decision Processes*. Tech. rep. arXiv, Feb. 2015. DOI: 10.48550/arXiv.1502.02259. arXiv: 1502.02259.

[15] Hado van Hasselt, Arthur Guez, and David Silver. "Deep Reinforcement Learning with Double Q-Learning". In: *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. AAAI'16. Phoenix, Arizona: AAAI Press, Feb. 2016, pp. 2094–2100.

[16] Eric Jang, Shixiang Gu, and Ben Poole. "Categorical Reparameterization with Gumbel-Softmax". In: *International Conference on Learning Representations*. 2017. URL: https://openreview.net/forum?id=rkE3y85ee.

[17] Melvin Laux et al. *Contextual Multi-Task Reinforcement Learning for Autonomous Reef Monitoring*. 2026. arXiv: 2604.12645 [cs.RO]. URL: https://arxiv.org/abs/2604.12645.

[18] Michael A. Lepori, Thomas Serre, and Ellie Pavlick. "Break It Down: Evidence for Structural Compositionality in Neural Networks". In: *Thirty-seventh Conference on Neural Information Processing Systems*. 2023. URL: https://openreview.net/forum?id=rwbzMiuFQl.

[19] Eran Malach et al. "Proving the Lottery Ticket Hypothesis: Pruning is All You Need". In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, 13–18 Jul 2020, pp. 6682–6691. URL: https://proceedings.mlr.press/v119/malach20a.html.

[20] Stephanie Milani et al. "Explainable Reinforcement Learning: A Survey and Comparative Review". In: *ACM Comput. Surv.* 56.7 (Apr. 9, 2024), 168:1–168:36. ISSN: 0360-0300. DOI: 10.1145/3616864. URL: https://dl.acm.org/doi/10.1145/3616864.

[21] Aditya Modi et al. "Markov Decision Processes with Continuous Side Information". In: *Algorithmic Learning Theory, ALT 2018, 7-9 April 2018, Lanzarote, Canary Islands, Spain*. Ed. by Firdaus Janoos, Mehryar Mohri, and Karthik Sridharan. Vol. 83. Proceedings of Machine Learning Research. PMLR, 2018, pp. 597–618.

[22] Easton Potokar et al. "HoloOcean: A Full-Featured Marine Robotics Simulator for Perception and Autonomy". In: *IEEE Journal of Oceanic Engineering* 49.4 (Oct. 2024), pp. 1322–1336. ISSN: 1558-1691. DOI: 10.1109/JOE.2024.3410290. URL: https://ieeexplore.ieee.org/document/10638434.

[23] Jan Sobotka, Auke Ijspeert, and Guillaume Bellegarda. "Reverse-Engineering Memory in DreamerV3: From Sparse Representations to Functional Circuits". In: Mechanistic Interpretability Workshop at NeurIPS 2025. Sept. 30, 2025. URL: https://openreview.net/forum?id=JmjqTi4FDF.

[24] Tristan Trim and Triston Grayston. *Mechanistic Interpretability of Reinforcement Learning Agents*. Oct. 30, 2024. DOI: 10.48550/arXiv.2411.00867. arXiv: 2411.00867 [cs]. URL: http://arxiv.org/abs/2411.00867.

[25] Nelson Vithayathil Varghese et al. "A Survey of Multi-Task Deep Reinforcement Learning". In: *Electronics* 9.9 (Aug. 22, 2020). ISSN: 2079-9292. DOI: 10.3390/electronics9091363. URL: https://www.mdpi.com/2079-9292/9/9/1363.