* Look up example of title for lit review
* Add papers we have already to a list for ref and future adding
* Write title
* **Background research report (*1-2000 words*)**
* Your literature survey:
* justifies that your project is worth doing
* sets your work in context, critically evaluating past and current research
* provides a starting point for future work
* What inspires your work?
* This will most likely be a survey of similar projects, research articles, applications relevant to your project
* Engage critically with existing work.
* What are the boundaries, limitations, contradictions, developing areas, and dead ends of other work?
* Go beyond mere description by offering opinions to what is written
* How have other projects evaluated their work?
* Use this section to demonstrate an awareness of how your project fits within the context of the field(s) you are studying
* Provide consistently formatted references for all related work you discuss
* **Overall essentials**
* Implicitly, a good project proposal should:
* Introduce the subject area
* Highlight current research
* Identify a gap in current research
* Show how your work fills that gap
* Identify potential risks and solutions with the project

Who is it for – google algorithms and common use cases – who is it for

**A review of modern approaches to overcoming generalisation error in deep reinforcement learning.**

Abstract:   
In the domain of reinforcement learning, there are still many challenges to overcome on the journey to building agents with the capability to handle “real world scenarios, where the environment will be diverse, dynamic and unpredictable.” [1]

One of the biggest problems in the way of this goal is achieving a good degree of generalisation, where an agent is defined as able “generalise well to novel unseen situations” [2].   
This project will focus on the implementation of techniques aiming to improving the generalisation capability of an agent, primarily through the use of “procedural content generation (PCG), which seeks to algorithmically generate diverse levels.” [3]

By training over many varied procedurally generated levels, we aim to show improved capability to adapt to a new, unseen environment from training, avoid overfitting to the training environment.  
Experiments will be carried out in a small dungeon crawler-esque game environment, being a relatively approachable training ground with a discrete action and state space. We will then evaluate the performance of agents with increasing degrees of levels trained on. We will look at a baseline of no generalisation techniques, and then comparing the improvement in percentage of levels solved based on degree of training on PCG.

In the field of reinforcement learning (RL), significant challenges remain in developing agents capable of handling real-world scenarios, where environments are often diverse, dynamic, and unpredictable [1]. A primary obstacle in achieving this goal is enabling effective generalization, where an agent is defined as one that can "generalize well to novel, unseen situations" [2]. This project focuses on enhancing an agent's generalization capabilities, primarily through the use of **procedural content generation (PCG)**, which algorithmically generates diverse levels [3]. By training an agent on a wide range of procedurally generated environments, we aim to improve its ability to adapt to new, unseen scenarios, thus preventing overfitting to the training environment.

The experiments will be conducted in a small dungeon-crawler-like game, a relatively simple environment with a discrete action and state space, making it an accessible platform for testing. Agents will be evaluated based on their ability to solve levels after being trained on varying degrees of PCG-generated content. We will compare the performance of agents trained without generalization techniques to those trained on increasingly diverse PCG environments, measuring the improvement in the percentage of levels solved, time steps taken, and final score.

Intro:

Quick overview of machine learning:

Deeper dive into definition of reinforcement learning:

Highlight challenge of generalisation in new environments outside of training.  
Importance of generalisation.

Scope of review

Machine learning, existing under the umbrella of Artificial Intelligence, is a flourishing field in the world of computer science as we know it today. It was first defined in the 1950’s by a pioneer of Artificial Intelligence, Arthur Samuel, as “the field of study that gives computers the ability to learn without explicitly being programmed.” [4]  
But what does this really mean? How can a computer learn?  
The field of artificial intelligence has been through booms and ‘winters’ over the years, and in the 1990’s with the development of more powerful computers came a resurgence of machine learning. At this time, one machine learning application dominated the world, in the form of the spam-filter.  
In the year of 1997, Tom Mitchell defined the concept of Machine Learning with “A computer program is said to learn from experience E with respect to some task T and some performance measure P , if its performance on T , as measured by P , improves with experience E .” [5]  
This definition is perhaps more granular, and we can begin to see that the essence of machine learning comes from improving upon a particular task, through experience and evaluation of performance.  
Looking back at Tom Mitchell’s definition, a typical machine learning application might be to classify a class from an image or predict a numerical value, and receive a value of error (P) for their prediction (E) to then improve upon the task (T).  
This brings us to Reinforcement Learning, how is it different?  
  
Reinforcement learning is an exciting subfield of Machine Learning, which has also been around since the 1950’s, but with newfound popularity in the last decade due to the increased accessibility of deep learning.  
Reinforcement learning is of particular interest in the quest to solve complex, real world problems, as it is adaptive and looks at long term consequence of sequences of actions. [7]  
In Reinforcement Learning, a software agent makes observations and takes actions within an environment, in doing so it receives rewards or penalties. It’s goal in this is to learn to behave in a way that will maximize its expected rewards and minimise penalties over time. [6]  
Once again looking at Tom Mitchell’s definition, Our software agent in this instance learns from evaluation of the rewards (P) garnered from its actions within an environment (E) and improves it’s performance on a task (T) through this evaluation of experience.

This is all well and good, and we can see impressive results in many applications, from besting human performance in complex games with continuous action spaces such as DoTA [8], the Chinese game of Go with its enormous search space and difficult in evaluating moves and board state [9], or more real world applications such as autonomous driving [10, 11].

But reinforcement learning still faces many challenges in the face of achieving solutions to the complex and dynamic nature of real world problems, and one of the largest problems that the field is actively researching to solve is that of generalisation.

1. **Generalisation**

When we talk about the ability of an agent to generalise, we mean the ability to generalise learning from previously experienced states [7] to ones that are novel and unseen. [2]

Reinforcement learning agents are prone to overfitting their environments, [13] and this leads to an inability to deal with new situations. This is especially important to tackle, as the real world is dynamic and complex, and we need agents able to deal with diverse situations and not fail catastrophically in the face of the unseen.

The scope of this this review will be that of modern techniques used to aid in the improvement and evaluation of generalisation capability of a software agent.  
These techniques vary from the augmentation of data and regularisation of models, to training across varied and diverse procedurally generated environments in an attempt to train underlying behaviours rather than overfit one environment.

Overview of Generalisation in DRL:

Define generalisation more formally, discuss related concepts (domain adaptation, transfer learning, etc)

Generalisation is similar in concept to transfer learning and domain adaptation, which is a specific type of transfer learning.  
In the case of transfer learning, the goal is to leverage knowledge from a previously learned task to learn a new target task quicker than if there was no transfer at all. [14] In this case, we are taking knowledge from one task to a different task, for example from one game to another.  
Domain adaptation is a specific type of transfer learning, where the task may remain relatively the same, but we are dealing with domain shifts, such as from a simulation to the real world, and aiming to achieve meaningful generalisation to this other setting. [15]

In the case of generalisation, we are staying within the same setting, and looking for a robust performance on unseen instances of the same task.

This task is not an easy one, and there is a great chance that a model will still overfit to a considerable, even when using techniques to combat this, as in the case of the CoinRun benchmark (Cobbe et al. 2019) [16].

Key techniques of improving generalisation in DRL

Data augmentation

Batch normalisation

Regularisation

PCG

Split into subsections and discuss pros and cons

IDENTIFYING A GAP – Which algorithms and techniques?  
What has been done:  
List papers:

1. coinrun
2. PCG / PPCG : we show that this can lead to overfitting on a higher level, such as the distribution of generated levels presented during training.
3. ACCEL
4. Scan ZSG
5. Stochasticity exploration boltzmann

What are the limitations faced?

1. Coinrun – large scale timesteps, still overfitting despite lots of stuff
2. Difficulty in finding the right reward schemes, hyper parameters
3. PCG can lead to other types of overfitting

Dead ends? Boundaries? PCG is not a cure-all solution alone (ZSG) paper Developing areas? ACCEL, SCAN ZSG, CRL

What is my opinion of what has been written

How were those projects evaluated

dHRL look at total score, coin run looks at levels solved

generalisation error

timesteps avg

**What is the gap I will focus on:**  
dHRL recreation with generalisation, compared with techniques on a simple DQN

**How does this fit with the current field?**

dHRL is a novel algorithm

What is MaxQ

Why dHRL

dHRL has not been tested with generalisation

comparison to a DQN and other methods of ZSG

**Identify potential risks and pitfalls with my work**

I have no clue what im doing

Computational limits perhaps

My brains computational limits also

Overfitting despite generalisation techniques

Reproducibility of a novel algorithm  
  
REFERENCES:

**[1][2]** R. Kirk, A. Zhang, E. Grefenstette, and T. Rocktäschel, ‘A Survey of Zero-shot Generalisation in Deep Reinforcement Learning’, *jair*, vol. 76, pp. 201–264, Jan. 2023, doi: [10.1613/jair.1.14174](https://doi.org/10.1613/jair.1.14174).  
**[3]** A. A. J. P.-H. O. U. M. J. UCL, M. A. M. D. U. B. M. S. UCL, M. A. J. F. O. U. E. G. UCL, M. A. T. R. UCL, M. A. \*Equal contribution P. March 3, and 2022 ARXIV Full paper, ‘Evolving Curricula’. Accessed: Nov. 08, 2024. [Online]. Available:<https://accelagent.github.io>

**[4]** ‘Machine learning, explained | MIT Sloan’. Accessed: Nov. 13, 2024. [Online]. Available: <https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained>

**[5][6]** A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. Sebastopol, UNITED STATES: O’Reilly Media, Incorporated, 2019. Accessed: Nov. 13, 2024. [Online]. Available: <http://ebookcentral.proquest.com/lib/goldsmiths/detail.action?docID=5892320>

**[7]** R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, Massachusetts London, England: The MIT Press, 2015. Accessed: Nov. 13, 2024. [Online]. Available: <https://web.stanford.edu/class/psych209/Readings/SuttonBartoIPRLBook2ndEd.pdf>

**[8]** OpenAI *et al.*, ‘Dota 2 with Large Scale Deep Reinforcement Learning’, Dec. 13, 2019, *arXiv*: arXiv:1912.06680. doi: [10.48550/arXiv.1912.06680](https://doi.org/10.48550/arXiv.1912.06680).

**[9]** D. Silver *et al.*, ‘Mastering the game of Go with deep neural networks and tree search’, *Nature*, vol. 529, no. 7587, pp. 484–489, Jan. 2016, doi: [10.1038/nature16961](https://doi.org/10.1038/nature16961).

**[10]** S. Wang, D. Jia, and X. Weng, ‘Deep Reinforcement Learning for Autonomous Driving’, May 19, 2019, *arXiv*: arXiv:1811.11329. doi: [10.48550/arXiv.1811.11329](https://doi.org/10.48550/arXiv.1811.11329).

**[11]** J. Levinson et al., ‘Towards fully autonomous driving: Systems and algorithms’, in 2011 IEEE Intelligent Vehicles Symposium (IV), Jun. 2011, pp. 163–168. doi: 10.1109/IVS.2011.5940562.

**[13]** C. Zhao, O. Sigaud, F. Stulp, and T. M. Hospedales, ‘Investigating Generalisation in Continuous Deep Reinforcement Learning’, Feb. 20, 2019, *arXiv*: arXiv:1902.07015. doi: [10.48550/arXiv.1902.07015](https://doi.org/10.48550/arXiv.1902.07015).

**[14]** M. E. Taylor and P. Stone, ‘Transfer Learning for Reinforcement Learning Domains: A Survey’, *Journal of Machine Learning Research*, vol. 10, no. 56, pp. 1633–1685, 2009.

**[15]** J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, ‘Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World’, Mar. 20, 2017, *arXiv*: arXiv:1703.06907. Accessed: Nov. 13, 2024. [Online]. Available: <http://arxiv.org/abs/1703.06907>

**[16]** K. Cobbe, O. Klimov, C. Hesse, T. Kim, and J. Schulman, ‘Quantifying Generalization in Reinforcement Learning’, Jul. 14, 2019, *arXiv*: arXiv:1812.02341. Accessed: Nov. 10, 2024. [Online]. Available: <http://arxiv.org/abs/1812.02341>

**[17]** N. Justesen, R. R. Torrado, P. Bontrager, A. Khalifa, J. Togelius, and S. Risi, ‘Illuminating Generalization in Deep Reinforcement Learning through Procedural Level Generation’, Nov. 29, 2018, *arXiv*: arXiv:1806.10729. doi: [10.48550/arXiv.1806.10729](https://doi.org/10.48550/arXiv.1806.10729).

LIT REVIEW PAPER NOTES

Prime candidate, a novel dHRL system.

Paper = Hierarchical Reinforcement Learning for Playing a Dynamic Dungeon Crawler Game

<https://ieeexplore.ieee.org/document/8628914>

**Paper looking at the performance of a dHRL algorithm vs MaxQ-Q on a simple dungeon crawler setting.  
Mentions future work ideas of generalisation.**

Algorithm: dHRL, maxQ-Q

Training techniques: Boltzmann exploration, MLPs

“During training both start with a temperature of 4 for the Boltzmann exploration algorithm which is multiplied by 0.98 after each game, to a minimum of 0.1. Their neural network learning rates are initialised slightly differently, for MaxQ-Q it starts at 0.001 while for dHRL it starts at 0.0005. Both have a decay after each game of 0.995.”  
“We found that when a greedy policy is used during testing, both AIs sometimes got stuck in small loops of states. When the Boltzmann exploration algorithm is used with a low temperature, it significantly improves performance for both systems. Hence both use the Boltzmann algorithm with a temperature of 0.1 during performance tests.”Evaluation: Total score from task, standard error Score / Epochs

Avg Score / Mean error/ Win rate

Paper 2 =Quantifying Generalization in Reinforcement Learning

<https://arxiv.org/pdf/1812.02341>

**OpenAI paper looking at procedural level generation and the generalisation error in a 2d platformer Coinrun.**

**It talks about the surprising level of generalisation error, techniques to measure this level of overfitting in generalisation, and ways to reduce it through convolutional architectures and regularisation.**

Algorithm: Nature-CNN, IMPALA-CNN, PPO

Training techniques: PCG, Dropout, L2 Regularisation, Data Augmentation, Stochasticity (environmental through epsilon greedy action selection, Batch Normalization

Notes: “In previous work, epsilon-greedy action selection has been used both as a means to encourage exploration and as a theoretical safeguard against overfitting (Bellemare et al., 2012; Mnih et al., 2013).”

“We find that combining data augmentation, batch normalization, and L2 regularization yields slightly better test time performance than using any one of them individually.”

“We have observed the surprising extent to which agents can overfit to a fixed training set. Using the procedurally generated CoinRun environment, we can precisely quantify such overfitting. With this metric, we can better evaluate key architectural and algorithmic decisions. We believe that the lessons learned from this environment will apply in more complex settings, and we hope to use this benchmark, and others like it, to iterate towards more generalizable agents.”

ON PCG “Mazes are generated using Kruskal’s algorithm (Kruskal, 1956).”

ON L2 and DO “They turn to supervised learning for inspiration, finding that both L2 regularization and dropout can help agents learn more generalizable features.”

Evaluation: Generalisation error through Levels Solved% of train vs test on number of training levels, levels solved% of certain levels solved based on number of timesteps

Paper3= NETWORK RANDOMIZATION:

A SIMPLE TECHNIQUE FOR GENERALIZATION

IN DEEP REINFORCEMENT LEARNING <https://arxiv.org/pdf/1910.05396>

**This paper talks about improving generalisation capabilities through the addition of a CNN to randomise the rendering inputs to a DRL to help with generalisation, and sees success.**

Published as a conference paper at ICLR 2020

Algorithm: PPO, IMPALA-CNN

Training Techniques: CNN Network Randomisation, Dropout, L2 Regularisation, Data augmentation

Notes: **REFERENCES** “However, it has been evidenced in recent years that deep RL agents of-

ten struggle to generalize to new environments, even when semantically similar to trained agents

(Farebrother et al., 2018; Zhang et al., 2018b; Gamrian & Goldberg, 2019; Cobbe et al., 2019). For

example, RL agents that learned a near-optimal policy for training levels in a video game fail to

perform accurately in unseen levels (Cobbe et al., 2019), while a human can seamlessly generalize

across similar tasks. Namely, RL agents often overfit to training environments, thus the lack of gen-

eralization ability makes them unreliable in several applications, such as health care (Chakraborty

& Murphy, 2014) and finance (Deng et al., 2016)”

**Failure of agents to generalise on small visual changes** : “We also found that RL agents completely fail

due to small visual changes1 because it is challenging to learn generalizable representations from

high-dimensional input observations, such as images.”  
**Characterisation of Generalisation**: “The generalization of RL agents can be characterized by visual changes (Cobbe et al., 2019; Gam-

rian & Goldberg, 2019), different dynamics (Packer et al., 2018), and various structures (Beattie

et al., 2016; Wang et al., 2016).”

**On methods of improving generalisation:** “To improve generalization, several strategies, such as regularization (Farebrother et al., 2018; Zhang

et al., 2018b; Cobbe et al., 2019) and data augmentation (Tobin et al., 2017; Ren et al., 2019),

have been proposed in the literature (see Section 2 for further details). In particular, Tobin et al.

(2017) showed that training RL agents in various environments generated by randomizing rendering

in a simulator improves the generalization performance, leading to a better performance in real

environments.”

**CNN Network Randomisation:** “Our main idea is to

utilize random (convolutional) networks to generate randomized inputs (see Figure 1(a)), and train

RL agents (or their policy) by feeding them into the networks. Specifically, by re-initializing the

parameters of random networks at every iteration, the agents are encouraged to be trained under a

broad range of perturbed low-level features, e.g., various textures, colors, or shapes.”

Paper 4= **This paper looks at generalisation and overfitting using procedural level generation for 4 different games.**

Illuminating Generalization in Deep Reinforcement Learning

through Procedural Level Generation

<https://rlg.mlanctot.info/papers/AAAI19-RLG-Paper22.pdf>

Algorithms: A2C

Training Techniques: **GVG AI (environment),** **Progressive PCG, RMS Optimizer**

Notes: **On architecture of their A2C** “The neural networks in this paper have the same architecture originally used from Mnih et al. (Mnih et al. 2016) with three convolutional layers and a single fully-connected layer.”

**On curricula or increasing difficult generation of content** “Addi-

tionally, it is possible to achieve better performance with less

data by manipulating the difficulty of the levels in response

to the performance of the agent.”

“Progressive PCG,

where the difficulty of levels/tasks is increased gradually to

match the agent’s performance. While similar techniques of

increasing difficulty have been used before, they have not

been combined with a PCG-based approach in which agents

are evaluated on a completely new level every time a new

episode begins.”

Evaluation: Score/Steps/Difficulty

Max score vs random baseline score vs actual score

Paper 5: **Deep Reinforcement Learning for General Video Game AI**

<https://ar5iv.labs.arxiv.org/html/1806.02448>

Algorithms: DQN, Dueling DQN, A2C

Training Techniques: GVGAI (environment)

Notes: **On gvgai** “The General Video Game AI (GVGAI) competitions and framework were created with the express purpose of providing a versatile general AI benchmark *[*[*3*](https://ar5iv.labs.arxiv.org/html/1806.02448#bib.bib3)*,* [*4*](https://ar5iv.labs.arxiv.org/html/1806.02448#bib.bib4)*,* [*5*](https://ar5iv.labs.arxiv.org/html/1806.02448#bib.bib5)*,* [*6*](https://ar5iv.labs.arxiv.org/html/1806.02448#bib.bib6)*]*. The *planning tracks* of the competition, where agents are given a forward model allowing them to plan but no training time between games, have been very popular and seen a number of strong agents based on tree search or evolutionary planning submitted.”

**EVALUATION:** Mean episode reward / timesteps taken

**On evaluation** “This is done by reporting the sum of the incremental rewards for the episode at a given time step. Since this data is noisy due to episode restarts, the 20 results are averaged to smooth the graph and better show a trend. A2C allows running in parallel, we were able to run 12 networks in parallel at once. To keep the comparisons fair, A2C is still only allowed one million GVGAI calls and therefore each of the 12 networks is given one-twelfth of a million calls each. This results in the training graph seen in Figure [4](https://ar5iv.labs.arxiv.org/html/1806.02448#S4.F4). To compare this with the linear algorithms, each time step of A2C is associated with 12 time-steps of the DQN algorithms in Figure [3](https://ar5iv.labs.arxiv.org/html/1806.02448#S4.F3). The value for each time step of A2C is the average of all 12 rewards.”

PAPER 6 =[1] Evolving Curricula with Regret-Based Environment Design

A. A. J. P.-H. O. U. M. J. UCL, M. A. M. D. U. B. M. S. UCL, M. A. J. F. O. U. E. G. UCL, M. A. T. R. UCL, M. A. \*Equal contribution P. March 3, and 2022 ARXIV Full paper, ‘Evolving Curricula’. Accessed: Nov. 08, 2024. [Online]. Available: <https://accelagent.github.io>

**Paper looks at training a generalised model using curricula, teacher and a student models with good results**

**Also mentions Prioritized Level Replay**

Algorithms: ACCEL, PAIRED, POET, PLR

Training Techniques:UED

Notes: **On efficacy of DR** “However, DR (domain randomisation) is often not enough to train robust agents in domains where the agent struggles to make progress on many challenging levels.”

**On the realism of an MDP framework** “Despite the generality of the MDP framework, it is often an unrealistic model for real-world environments. First, it assumes full observability of the state, which is often impossible in practice. This is addressed in partially-observable MDPs, or POMDPs, which include an observation function \mathcal{I}: S \rightarrow O which maps the true state (which is unknown to the agent) to a potentially noisy set of observations O. Secondly, the traditional MDP framework assumes a single reward and transition function, which are fixed throughout learning. Instead, in the real world, agents may experience variations not seen during training, which makes it crucial that policies are capable of robust transfer.”

**On Dr**: “Domain Randomization (DR) can be viewed as the most basic form of UED.”

**On PCG frameworks:** “Our work also relates to the field of procedural content generation (PCG), which seeks to algorithmically generate diverse levels. Popular PCG environments used in RL include the Procgen Benchmark, MiniGrid, Obstacle Tower, GVGAI, and the NetHack Learning Environment. This work uses the recently proposed MiniHack environment, which provides a flexible framework for creating diverse environments.”

**On regret based algorithms:** “Dennis et al, 2020 first formalized UED and introduced the PAIRED algorithm, a minimax regret UED algorithm whereby an environment adversary learns to present levels that maximize regret, approximated as the difference in performance between the main student agent and a second agent.”

Evaluation: Test Return (score from level)

**This paper talks about the difficulty of avoiding memorisation of data in a RL setting, particularly with a continuous domain.**

**It also offers some techniques to avoid it.**

A Dissection of Overfitting and Generalization in

Continuous Reinforcement Learning

<https://arxiv.org/pdf/1806.07937>

**This paper surveys recent research of zero shot generalisation in DRL, aiming to improve algorithms capable of adapting to real world settings.  
They also aim to present a framework classifying ZSG as a class of problems, and state that generalisation suffers from the no free lunch theorem, and thus is not one singular problem.**

**Doing this they categorise types of generalisation.**

This paper surveys recent research on zero-shot generalization (ZSG) in deep reinforcement learning (RL), aiming to create algorithms capable of robust, adaptable performance in real-world settings. The authors present a unifying framework that formalizes ZSG as a class of problems, rather than a single one, to help researchers better understand and address its complexity. They categorize types of generalization, such as combinatorial and interpolation versus extrapolation, and clarify the importance of context awareness in training. The paper reviews existing benchmarks, suggests underexplored problem areas, and provides guidance for future research to foster progress toward more universally generalizable RL methods.

“We recommend that future environments should use a combination of PCG and controllable factors of variation.”

A Survey of Zero-shot Generalisation in Deep Reinforcement Learning

<https://jair.org/index.php/jair/article/view/14174/26890>

**ON difficulties in evaluating generalisation** “This means using it as theonly metric for improved performance will likely not lead to robust progress in ZSG. Further,given how broad the current set of assumptions is, it is unlikely there is a single generalmeasure of progress towards tackling ZSG: across such a broad problem class, objectivesmay even be conflicting (Wolpert & Macready, 1997).”