How Ensembles of Distilled Policies Improve Generalisation in Reinforcement Learning

Max Weltevrede
Delft University of Technology
Delft, The Netherlands
m.r.weltevrede@tudelft.nl

Moritz A. Zanger
Delft University of Technology
Delft, The Netherlands
m.a.zanger@tudelft.nl

Matthijs T. J. Spaan
Delft University of Technology
Delft, The Netherlands
m.t.j.spaan@tudelft.nl

Wendelin Böhmer
Delft University of Technology
Delft, The Netherlands
j.w.bohmer@tudelft.nl

Abstract

In the zero-shot policy transfer setting in reinforcement learning, the goal is to train an agent on a fixed set of training environments so that it can generalise to similar, but unseen, testing environments. Previous work has shown that policy distillation after training can sometimes produce a policy that outperforms the original in the testing environments. However, it is not yet entirely clear why that is, or what data should be used to distil the policy. In this paper, we prove, under certain assumptions, a generalisation bound for policy distillation after training. The theory provides two practical insights: for improved generalisation, you should 1) train an ensemble of distilled policies, and 2) distil it on as much data from the training environments as possible. We empirically verify that these insights hold in more general settings, when the assumptions required for the theory no longer hold. Finally, we demonstrate that an ensemble of policies distilled on a diverse dataset can generalise significantly better than the original agent.

1 Introduction

A major challenge for developing reliable reinforcement learning (RL) agents is their ability to generalise to new scenarios they did not encounter during training. The zero-shot policy transfer setting (ZSPT, Kirk et al., 2023) tests for this ability by having an agent train on a fixed set of training environments, referred to as training *contexts*, and measuring the agent's performance on a held-out set of similar, but different, testing contexts. Previous work has identified that *policy distillation* after training, the act of transferring knowledge from the agent's policy into a freshly initialised neural network, can be used as a tool for generalisation. In particular, it has been shown that the distilled policy sometimes achieves higher test performance than the original policy (Lyle et al., 2022).

However, it is not yet entirely clear *how* policy distillation after training can improve generalisation performance in RL. Lyle et al. (2022) theoretically show that the temporal difference (TD) loss negatively affects the smoothness of the learned value function, which only indirectly explains why policy distillation after training (without TD loss) can improve generalisation. They also partially attribute the observed generalisation benefits to the stationarity of the distillation targets, which avoids negative effects induced by the non-stationary RL targets during training (Igl et al., 2021), but this lacks a solid theoretical justification. Moreover, recent work has shown that not only the stationarity of the RL training distribution, but also its overall diversity can affect generalisation to unseen contexts (Jiang et al., 2023; Suau et al., 2024; Weltevrede et al., 2025). This additionally

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

raises the question whether we can increase generalisation performance by changing the distribution of data on which the policy is distilled.

In this paper, we theoretically analyse the act of distilling a policy after training, and try to answer *how* the policy should be distilled and on *what* data. Our analysis is based on the idea that many real-world data distributions exhibit symmetries, and that generalising to novel inputs will require being invariant to those symmetries. Although there is a lot of empirical evidence that demonstrates neural networks can learn invariances from data in a wide variety of settings and applications (Shorten and Khoshgoftaar, 2019; Feng et al., 2021; Zhang et al., 2021), proving this often requires stricter assumptions. Therefore, we analyse policy distillation in a *generalisation through invariance* ZSPT (GTI-ZSPT) setting, in which the agent has to learn invariance to a symmetry group, whilst only observing a subgroup of those symmetries during training. For this setting, we prove a generalisation bound for a distilled policy, and deduce insights that should translate beyond the strict group theoretical framework required for the theory.

Specifically, our theoretical results lead to two practical insights: generalisation performance can be improved by 1) training an ensemble of distilled policies, and 2) distilling on a diverse set of states. Training an ensemble can be very costly. However, we demonstrate that generalisation can be improved (at only a fraction of the sample cost required for training the RL agent), by instead creating an ensemble *after* training by distilling the agent several times and averaging the resulting policy. Finally, related to our work on policy distillation, recent work has suggested that the generalisation performance of behaviour cloning (BC) is competitive with state-of-the-art offline RL in the ZSPT setting (Mediratta et al., 2024). We demonstrate the insights for policy distillation also transfer to the BC setting and produce better generalising behaviour cloned policies. Our contributions are:

- Given a policy (for example, an RL agent after training), we prove a bound on the test performance for a distilled policy in the GTI-ZSPT setting. This bound is improved by 1) distilling a larger ensemble of policies, and 2) distillation over a more diverse set of states.
- Inspired by the theoretical results, we empirically show that the insights gained from the theory improve generalisation of behaviour cloned and distilled policies in more general settings, when the strict assumptions required for the theory no longer hold. Furthermore, we demonstrate that an ensemble of policies distilled on a diverse dataset can generalise significantly better than the original RL agent.

2 Background

The goal in reinforcement learning is to optimise a decision-making process, usually formalised as a Markov decision-making process (MDP) defined by the 6 tuple $\mathcal{M} = (S, A, T, R, p_0, \gamma)$. In this tuple, S denotes the state space, A the action space, $T : S \times A \to \mathcal{P}(S)$ the transition model, $R : S \times A \to \mathbb{R}$ the reward function, $p_0 : \mathcal{P}(S)$ the initial state distribution and $\gamma \in [0, 1]$ the discount factor, where $\mathcal{P}(S)$ denotes the probability function over state space S. Optimising an MDP corresponds to finding the policy $\pi : S \to \mathcal{P}(A)$ that maximises the return (the expected discounted sum of rewards) $J^\pi = \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r_t]$. The expectation here is over the Markov chain $\{s_0, a_0, r_0, s_1, a_1, r_1, ...\}$ induced by following policy π in MDP $\mathcal{M}$ (Akshay et al., 2013). The optimal policy $\pi^* = \arg\max_\pi \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r_t]$ is the policy that maximises this return. The *on-policy* distribution $\rho^\pi_\mathcal{M} : \mathcal{P}(S)$ is the distribution over the states that a policy π would visit in an MDP $\mathcal{M}$.

A contextual Markov decision-making process (CMDP) $\mathcal{M}|_C$ (Hallak et al., 2015) is an MDP where the state space $S = S' \times C$ can be structured as an outer product of a context space C and underlying state space S'. A context $c \in C$ is sampled at the start of an episode and does not change thereafter. The context is part of the state and can influence the transitions and rewards. As such, it can be thought of as defining a task or specific environment that the agent has to solve in that episode. In the *zero-shot policy transfer* (ZSPT) setting (Kirk et al., 2023), an agent gets to train on a fixed subset of contexts $C_{train} \subset C$ and has to generalise to a distinct set of testing contexts $C_{test} \subset C, C_{train} \cap C_{test} = \varnothing$. In other words, the agent gets to interact and train in the CMDP $\mathcal{M}|_{C_{train}}$ (the CMDP induced by the training contexts C_{train}), but has to maximise return in the testing CMDP $\mathcal{M}|_{C_{test}}$.

2.1 Policy distillation

In policy distillation, a knowledge transfer occurs by distilling a policy from a *teacher* network into a newly initialised *student* network. There are many different ways the policy can be distilled

(Czarnecki et al., 2019), but in this paper we consider a student network that is distilled on a fixed dataset, that is collected after training, and usually (but not necessarily) consists of on-policy data collected by the teacher. For analysis, we simplify the setting by assuming a deterministic, scalar student and teacher policy $\pi_\theta : S \to \mathbb{R}, \pi_\beta : S \to \mathbb{R}$. The distillation loss we consider is simply the mean squared error (MSE) between the output of the two policies:

$$l_D(\theta, \mathcal{D}, \pi_\beta) = \frac{1}{n} \sum_{s \in \mathcal{D}} (\pi_\theta(s) - \pi_\beta(s))^2 \tag{1}$$

where $\mathcal{D} = \{s_1, ..., s_n\}$ is the set of states we distil on. This simplified distillation setting is only used for the theoretical results, our experiments in Section 5 consider more general settings. Note, we consider *behaviour cloning* (BC) as a specific instance of policy distillation, where the student network only has access to a fixed dataset of the teacher's behaviour (state-action tuples). For more on behaviour cloning, distillation and their differences, we refer to Appendix A.1.

If we assume a certain smoothness of the transitions, rewards and policies (in particular, Lipschitz continuous MDP and policies), it is possible to bound the performance difference between the student and an optimal policy (Maran et al., 2023, Theorem 3):

Theorem 3. *Let π^* be the optimal policy and π_θ be the student policy. If the MDP is (L_T, L_R)-Lipschitz continuous and the optimal and student policies are L_π-Lipschitz continuous, and we have that $\gamma L_T(1 + L_\pi) < 1$, then it holds that:*

$$J^{\pi^*} - J^{\pi_\theta} \leq \frac{L_R}{(1-\gamma)(1-\gamma L_T(1+L_{\pi_\theta}))} \mathbb{E}_{s \sim d^{\pi^*}}[\mathcal{W}(\pi^*(\cdot|s), \pi_\theta(\cdot|s))]$$

where $d^{\pi^*}(s) = (1-\gamma)\sum_{t=0}^{\infty} \gamma^t \mathbb{P}(s_t = s|\pi^*, p_0)$ the γ-discounted visitation distribution.

Proof. See Appendix E.1 for the proof and exact definitions of all the terms. □

In other words, under these conditions the return of a student policy can be bounded by the distance between the student and optimal policies along the states visited by the optimal policy.

2.2 Symmetry groups

To formalise the notion of symmetries and invariance of a function $f : X \to Y$, it is useful to define a *symmetry group* G. A group is a non-empty set G together with a binary operation $\cdot$ that satisfies certain requirements such as closure, associativity and always containing an inverse and identity element.[1] A group and its elements are abstract mathematical notions. In order to apply elements from a group to a vector space X, we need to define a *group representation* ψ_X that maps group elements to invertible matrices. In this paper, we always assume the representations are orthogonal (i.e. $\psi_X(g^{-1}) = \psi_X(g)^\top$). Note that we only need to define the representations ψ_X for the analysis, as they are part of the generalisation bound but are not explicitly defined for the experiments.

We can define the invariance of a function f as

$$f(\psi_X(g)x) = f(x) \quad \forall x \in X, g \in G. \tag{2}$$

A subset B of G is called a *subgroup* of G (notation $B \leq G$) if the members of B form a group themselves. Any group G has at least one subgroup, the *trivial subgroup*, consisting of only the identity element $e : e \circ g = g \circ e = g$, $\forall g \in G$. A subgroup $B \leq G$ is finite if it has a finite number of elements. For more on group theory, we refer to Appendix A.2.

Example The group $SO(2)$ consists of the set of all rotations in two dimensions. If the input to function f consists of Euclidean coordinates (x, y), the group representation ψ_X maps a rotation of α degrees to the 2D rotation matrix associated with an α degree rotation. The function f would be considered rotationally invariant if $f(\psi_X(\alpha)x) = f(x)$, $\forall x \in \mathbb{R}^2, \alpha \in SO(2)$. An example of a finite subgroup of $SO(2)$ is the group C_4 consisting of all $90°$ rotations, or the subgroup consisting of only the identity element ($0°$ rotation).

[1] In this paper, we abuse notation slightly by denoting both the group and the non-empty set with G, depending on context.

One approach to induce a function that is invariant to the symmetry group G is to train it with *data augmentation*. For groups of finite size, it is possible to perform *full data augmentation*, which consists of applying every transformation in G, to each element of an original dataset $\mathcal{T} = (\mathcal{X}, \mathcal{Y}) = \{(x, y) \in X, Y\}^n$. The function is then trained on the *augmented* dataset $\mathcal{T}_G = \{(\psi_X(g)x, y) \big| \forall (x, y) \in \mathcal{T}, g \in G\}$. In general, training a function with data augmentation does not guarantee it becomes invariant (Flinth and Ohlsson, 2023), it instead can become approximately invariant or invariant only on the distribution of data on which it was trained (Kvinge et al., 2022; Lyle et al., 2020; Azulay and Weiss, 2019). However, under certain conditions, the average of an infinitely large ensemble *can* have that guarantee (Gerken and Kessel, 2024; Nordenfors and Flinth, 2024).

2.3 Ensembles and invariance

Formally, an ensemble consists of multiple neural networks $f_\theta : X \to \mathbb{R}$ with parameters $\theta \sim \mu$ initialised from some distribution μ and trained on the same dataset $\mathcal{T} = (\mathcal{X}, \mathcal{Y}) = \{(x, y) \in X, Y\}^n$. The output of an infinitely large ensemble $\bar{f}_t(x)$ at training time t is given by the average over the *ensemble members* f_θ: $\bar{f}_t(x) = \mathbb{E}_{\theta\sim\mu}\big[f_{\mathcal{L}_t\theta}(x)\big]$, where $\mathcal{L}_t$ denotes a map from initial parameters θ to the corresponding parameters after t steps of gradient descent. In practice, the infinite ensemble is approximated with a finite Monte Carlo estimate of the expectation $\hat{f}_t$: $\bar{f}_t \approx \hat{f}_t = \frac{1}{N}\sum_{i=1}^{N} f_{\mathcal{L}_t\theta_i}(x)$, where $\theta_i \sim \mu$ and N is the size of the ensemble.

2.3.1 Infinite width limit

Although there does not yet exist a single comprehensive theoretical framework for how neural networks work, significant progress has been made in the field of deep learning theory in the limit of infinite layer width. In this limit, an infinite ensemble $\bar{f}_t(x)$ trained with MSE loss follows a simple Gaussian distribution that depends on the network architecture and initialisation (Jacot et al., 2018; Lee et al., 2019). Gerken and Kessel (2024) prove that the infinite ensemble $\bar{f}_t(x)$ trained on the augmented dataset $\mathcal{T}_G$ for some group G, satisfies the definition of invariance in equation (2), for any t and any x. In other words, an infinitely large ensemble of infinitely wide neural networks, trained with full data augmentation for group G, is invariant under transformations from G for any input and at any point during the training process. In our analysis, we use Lemma 6.2 from Gerken and Kessel (2024) that bounds the invariance of an infinite ensemble of infinitely wide networks trained with full data augmentation on a finite subgroup $B \leq G$:

Lemma 6.2. *Let $\bar{f}_t(x) = \mathbb{E}_{\theta\sim\mu}[f_{\mathcal{L}_t\theta}(x)]$ be an infinite ensemble of neural networks with Lipschitz continuous derivatives with respect to the parameters. Define the error κ as a measure of discrepancy between representations from the group G and its finite subgroup B:*

$$\kappa = \max_{g\in G} \min_{b\in B} ||\psi_X(g) - \psi_X(b)||_{op} \tag{3}$$

where $||\cdot||_{op}$ denotes the operator norm. The prediction of an infinite ensemble trained with full data augmentation on $B \leq G$ deviates from invariance by

$$\big|\bar{f}_t(x) - \bar{f}_t\big(\psi_X(g)x\big)\big| \leq \kappa\, C(x), \qquad \forall g \in G \tag{4}$$

for any time t. Here C is a function of x independent of g.

Proof. See Appendix E.2 for the proof and exact definitions of all the terms. □

This lemma bounds the deviation from invariance of the infinite ensemble by a factor κ, which is a measure of how well the subgroup B covers the full group G, in the space of representations ψ_X. For more background on infinite ensembles in the infinitely wide limit, we refer to Appendix A.3.

3 Related work

The CMDP framework captures many RL settings focused on zero-shot generalisation (Kirk et al., 2023). Some approaches to improve generalisation focus on learning generalisable functions through inductive biases (Kansky et al., 2017; Wang et al., 2021) or by applying regularisation techniques from supervised learning (Tishby and Zaslavsky, 2015; Cobbe et al., 2019). These approaches improve

generalisation by changing the RL training process, whereas we distil a teacher policy *after* training, which in principle is agnostic to how that teacher was trained. Other work improves generalisation by increasing the diversity of the data on which the agent trains, for example by increasing the diversity of the training contexts using domain randomisation (Tobin et al., 2017; Sadeghi and Levine, 2017), or creating artificial data using data augmentation (Lee et al., 2020; Raileanu et al., 2021). Our work focusses on sampling additional data from a fixed set of training contexts, but differs from data augmentation in that we do not require explicitly designed augmentations. For a broader survey on zero-shot generalisation in reinforcement learning, see Kirk et al. (2023).

Policy distillation in RL has been used to compress policies, speed up learning, or train multi-task agents by transferring knowledge from teacher policies to student networks (Rusu et al., 2016; Schmitt et al., 2018; Czarnecki et al., 2019). Various methods of distillation exist, balancing factors such as teacher quality, access to online data, and availability of teacher value functions or rewards (Czarnecki et al., 2019). Some studies have used distillation to improve generalisation, either by mitigating RL-specific non-stationarity through periodic distillation (Igl et al., 2021) or by distilling from policies trained with privileged information or weak augmentations (Fan et al., 2021; Walsman et al., 2023). Most similar to our work, Lyle et al. (2022) show a policy distilled after training can sometimes generalise better than the original RL agent. But, their theory only indirectly covers policy distillation and they do not investigate how the distillation data affects generalisation.

4 Generalisation through invariance

In this section, we introduce a specific ZSPT setting that allows us to prove a generalisation bound for a distilled policy. The main idea is that many real-world data distributions exhibit symmetries, and that generalising to novel inputs sampled from this distribution requires (at least partially) being invariant to those symmetries. Moreover, any training dataset sampled IID from this distribution will likely observe some of these symmetries.

Proving a neural network learns invariances from data is not straightforward, and usually requires assumptions on the mathematical structure of the symmetries. For this reason, we consider a specific setting in which an agent has to become invariant to a symmetry group G, but trains with full data augmentation under only a subgroup $B \leq G$. Even though this setting requires strict assumptions, we expect the insights to apply more broadly, as there is a lot of empirical evidence that data augmentation improves generalisation performance in a wide variety of settings and applications (Shorten and Khoshgoftaar, 2019; Feng et al., 2021; Zhang et al., 2021; Miao et al., 2023). We formalise the idea in a *generalisation through invariance ZSPT* (GTI-ZSPT)

Definition 1 (Generalisation through invariance ZSPT)**.** *Let $\mathcal{M}|_C$ be a CMDP and let $C_{train}, C_{test} \subset C$ be a set of training and testing contexts that define a ZSPT problem. Additionally, let π^* be the optimal policy in $\mathcal{M}|_C$, $S^{\pi^*}_{\mathcal{M}|_C} = \{s \in S | \rho^{\pi^*}_{\mathcal{M}|_C}(s) > 0\}$ denote the set of states with non-zero support under the on-policy distribution $\rho^{\pi^*}_{\mathcal{M}|_C}$ in CMDP $\mathcal{M}|_C$. In the generalisation through invariance ZSPT (GTI-ZSPT), the sets $S^{\pi^*}_{\mathcal{M}|_C}$ and $S^{\pi^*}_{\mathcal{M}|_{C_{train}}}$ admit a symmetric structure:*

$$S^{\pi^*}_{\mathcal{M}|_C} = \{\psi_S(g)s | g \in G, s \in \bar{S}\}$$
$$S^{\pi^*}_{\mathcal{M}|_{C_{train}}} = \{\psi_S(b)s | b \in B, s \in \bar{S}\}, \quad B \leq G$$

where $\bar{S} \subset S^{\pi^*}_{\mathcal{M}|_{C_{train}}}$ is a proper subset of $S^{\pi^*}_{\mathcal{M}|_{C_{train}}}$ and G is a non-trivial symmetry group (and $B \leq G$ a finite subgroup) that leaves the optimal policy invariant: $\pi^*(s) = \pi^*(\psi_S(g)s), \forall s \in \bar{S}$.

To quantify the discrepancy between the group and its subgroup, the following measure is defined (Gerken and Kessel, 2024):

Definition 2. *For the group G and its finite subgroup $B \leq G$ that define the symmetric structure of a GTI-ZSPT (Definition 1), κ is a measure of discrepancy between the representations of these groups:*

$$\kappa = \max_{g \in G} \min_{b \in B} ||\psi_S(g) - \psi_S(b)||_{op}$$

where $|| \cdot ||_{op}$ denotes the operator norm.

The constant κ measures how much the subgroup $B \leq G$ deviates from the full group G. The bigger the subgroup B is, the smaller κ will become (with $\kappa = 0$ in the limit of $B = G$).

Example The 'Reacher with rotational symmetry' ZSPT in Figure 1 satisfies the conditions of the GTI-ZSPT. This is a continuous control environment where the agent has to move a robot arm (blue) in such a way that its hand (black circle) reaches the goal location (green circle). The four training contexts have shoulder locations that are rotated 0, 90, 180 and 270 degrees around the goal location. In testing, the shoulder can be rotated any amount. As an example, the measure κ for the subgroup C_4 of $90°$ rotations (as depicted in the figure), would be larger than for the bigger subgroup C_8 of $45°$ rotations. See Appendix C.1 for more on this example and how it satisfies the assumptions.

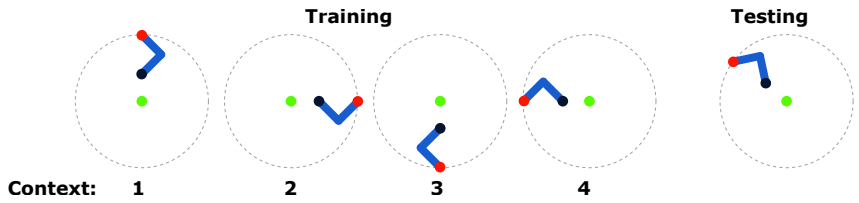


Figure 1: A 'Reacher with rotational symmetry' CMDP with four training contexts, differing in the location of the shoulder (red), positioned along a circle (dotted line). All contexts share the relative pose of the robot arm (blue). The goal is for the hand (black circle) to reach the goal location (green circle) in the middle. The training contexts can be generated by applying the group of $90°$ rotations to context 1, and the testing contexts can be generated with the full group of rotations ($SO(2)$).

4.1 Bounding the performance

For the GTI-ZSPT setting, we can bound the performance of a distilled policy in the testing CMDP with the following theorem:

Theorem 1. *Consider policy distillation for a deterministic, scalar teacher policy $\pi_\beta : S \to \mathbb{R}$ (Equation (1) in Section 2.1) in a L_T, L_R-Lipschitz continuous CMDP in the GTI-ZSPT setting. Let the student policy $\hat{\pi}_\infty$ be an ensemble of N infinitely wide neural networks $\pi_\theta : S \to \mathbb{R}$ with Lipschitz continuous derivatives with respect to its parameters, distilled on an on-policy dataset $\mathcal{D} = S^{\pi_\beta}_{\mathcal{M}|_{C_{train}}} = \{\psi_S(b)s | b \in B, s \in \bar{S}\}$ consisting of all the states in the training contexts encountered by the teacher in the GTI-ZSPT setting. Furthermore, let the student policy be $L_{\hat{\pi}_\infty}$-Lipschitz continuous and assume $\gamma L_T(1 + L_{\hat{\pi}_\infty}) < 1$.*

If the teacher is optimal in the training tasks C_{train} (but arbitrarily bad anywhere else), the performance of the student in the testing CMDP $\mathcal{M}|_{C_{test}}$ is bounded with probability at least $1 - \epsilon$, by:

$$J^{\pi^*} - J^{\hat{\pi}_\infty} \leq \frac{L_R}{(1-\gamma)(1-\gamma L_T(1+L_{\hat{\pi}_\infty}))} \left(\kappa \bar{C}_\Theta + \frac{1}{\sqrt{N}} \bar{C}_{\Sigma_\infty}(\epsilon) \right) \tag{5}$$

where κ is the measure of discrepancy between subgroup $B \leq G$ and full group G (see definition 2) and $\bar{C}_\Theta, \bar{C}_{\Sigma_\infty}$ are constants that depend on the γ-discounted visitation distribution of the optimal policy in $\mathcal{M}|_{C_{test}}$, the network architecture, and the dataset $\mathcal{D}$. Additionally, $\bar{C}_{\Sigma_\infty}$ also depends on the network initialisation and the confidence level ϵ.

Proof. Thanks to the symmetric structure of the GTI-ZSPT, we can bound the output of an infinite ensemble of distilled policies $\bar{\pi}_\infty$, when evaluated on testing states in $\mathcal{M}_{C_{test}}$, using the bound on the deviation from invariance from Section 2.3.1. This can be combined with a probabilistic bound for Monte Carlo estimators to bound the output of a finite ensemble $\hat{\pi}_\infty$ on the testing states. With this bound on the output of the student policy, we can use the performance bound for Lipschitz continuous MDPs from Section 2.1 to get our final result above. See Appendix B for the full proof. □

The theorem above offers two insights:

1. The bigger the ensemble size N, the smaller the bound on performance.
2. The bigger the subgroup B, the smaller the measure κ, the smaller the bound on performance.

As we mentioned before, even though Theorem 1 requires strict assumptions, we believe the insights apply more broadly. Essentially, the theorem relies on the generalisation benefits induced by training on additional samples generated by performing data augmentation. In practice, it often doesn't matter if the augmentations form a group, are consistent with the original data distribution, or are applied to all classes equally (Bishop, 1995; Wu et al., 2020; Hansen and Wang, 2021; Lin et al., 2022; Geiping et al., 2023; Miao et al., 2023). As such, we believe that in many settings, the benefits of training on a bigger subgroup B, can also be realised by simply training on more diverse data, which we clarify with some examples in our experiments.

5 Experiments

In this section, we demonstrate that the insights provided by the theory translate to practical and workable principles that can improve the generalisation performance of a distilled policy, beyond the performance of the original agent. In Section 5.1, we establish that bigger ensembles indeed improve generalisation and show what it means to train on a bigger subgroup $B \leq G$ in the illustrative CMDP from Figure 1. This experiment satisfies the assumptions for the GTI-ZSPT setting, but does not strictly satisfy some of the non-practical assumptions required for Theorem 1. In Section 5.2, we demonstrate that the insights also apply to the more complex Minigrid Four Rooms environment (Chevalier-Boisvert et al., 2023) that breaks most of the assumptions required for the proof in Section 4. For experimental details, see Appendix C.

5.1 Reacher with rotational symmetry

Table 1: Performance of distilled policies in the Illustrative CMDP from Figure 1 for different ensemble sizes N (trained under subgroup $B = C_4$) and different subgroups $B \leq SO(2)$ (for $N = 1$). Shown are the mean and standard deviation for 20 seeds, and in bold are the best returns including those with overlapping 95% confidence intervals.

Ensemble Size N:	N=1	N=10	N=100
Train Performance	**1.17 ± 0.004**	**1.17 ± 0.004**	**1.17 ± 0.003**
Test Performance	0.75 ± 0.147	0.89 ± 0.107	**1.05 ± 0.117**
Subgroup $B \leq SO(2)$:	$B = C_2$	$B = C_4$	$B = C_8$
Train Performance	**1.17 ± 0.003**	**1.17 ± 0.004**	1.16 ± 0.002
Test Performance	0.39 ± 0.0805	0.75 ± 0.147	**1.11 ± 0.072**

The theory proves that we can reduce an *upper bound* on the difference to optimal performance when we increase the ensemble size N and train on a bigger subgroup $B \leq G$. However, that does not always guarantee strict performance improvements (for example, if the upper bound were so large it is meaningless). Furthermore, the theory requires some assumptions that are not always practical, such as infinitely wide networks, scalar-valued policies, or a Lipschitz-continuous reward function, that we do not expect to affect the overall result in practice. Therefore, we investigate whether the insights from Section 4 hold without these assumptions, and whether they lead to actual generalisation improvements. In Table 1 we show that increasing the size of the ensemble, consisting of networks of finite width, *does* actually lead to higher test performance in the CMDP from Figure 1.

Additionally, in Table 1 we show that generalisation performance is affected by the size of the subgroup $B \leq SO(2)$ we train on. Only training on two training contexts, corresponding to the subgroup $C_2 \leq SO(2)$ of 180° rotations, performs worse than training on four contexts, corresponding to the subgroup $C_4 \leq SO(2)$ of 90° rotations (as shown in Figure 1). Furthermore, training on eight contexts (subgroup $C_8 \leq SO(2)$ of all 45° rotations) is even better. In this illustrative CMDP, training on larger subgroups requires training in new contexts, but this is not always the case.

5.1.1 Improving generalisation with diverse data from the same contexts

In sufficiently complex CMDPs, there are several dimensions of variation between different contexts. For example, we can add different starting poses to the contexts in the CMDP from Figure 1, such that a context is now defined by a rotation of the shoulder *and* the relative pose of the robot arm

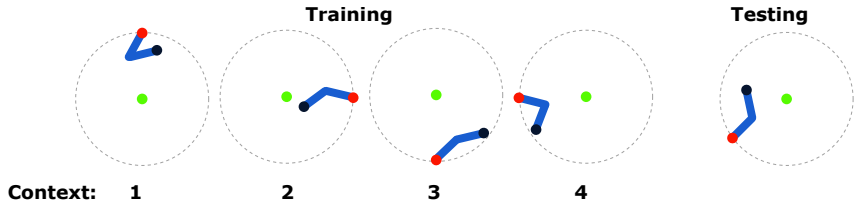


Figure 2: The base context set in the illustrative reacher CMDP with varying shoulder location (red) *and* robot arm pose (blue), see Figure 1 for details.

(see Figure 2). This CMDP does not strictly satisfy the symmetry conditions in Definition 1, but invariance to rotations is still a major component for generalisation. Since the training contexts now also differ in the starting pose, the dataset generated from the training contexts no longer corresponds to performing full data augmentation with respect to a rotational symmetry. However, in this example, this can be fixed by training on additional data from the given set of training contexts.

To illustrate this, we compare three distillation datasets:

1. **Training Contexts:** This dataset consists of the teacher's trajectories in the training contexts (contexts 1 through 4 in Figure 2).
2. **Training Contexts + C_4:** This dataset consists of the Training Contexts dataset, but with additional trajectories sampled from different starting poses in the same contexts. In particular, for each context, it includes trajectories starting in the rotated poses from the other contexts. This dataset corresponds to performing full data augmentation for the $90°$ rotations subgroup C_4 on the Training Contexts dataset (see Appendix C.1 for a visual representation of this).
3. **Training Contexts + Random:** Like Training Contexts + C_4, this dataset includes additional trajectories from different starting poses. However, for this dataset the new starting poses are sampled uniformly at random.

The Training Contexts + C_4 dataset illustrates how in this CMDP the subgroup $B \leq SO(2)$ can be increased by sampling additional trajectories from the same training contexts (technically, the Training Contexts dataset corresponds to the trivial subgroup $\{e\} \leq SO(2)$ consisting of only the identity element e, which is smaller than $C_4 \leq SO(2)$). In Table 2, we see that training on this dataset indeed produces higher test performance than the Training Contexts dataset. However, the same generalisation benefits are also observed for the Training Contexts + Random dataset.

Table 2: Performance of distilled policies (for $N = 1$) in the Illustrative CMDP from Figure 2 for different datasets. The datasets consist of the teacher's trajectories sampled for several starting states. Shown are the mean and standard deviation for 20 seeds, and in bold are the best returns including those with overlapping 95% confidence intervals.

Distillation Dataset	Train	Test
Training Contexts	**1.20 ± 0.157**	0.39 ± 0.051
Training Contexts + C_4	**1.11 ± 0.099**	**0.48 ± 0.080**
Training Contexts + Random	**1.14 ± 0.072**	**0.49 ± 0.077**

The Training Contexts + Random dataset illustrates that the generalisation benefits of data augmentation go far beyond the "training to be invariant under a group symmetry" paradigm. Some studies suggest that the benefits are simply due to the regularising effect that data augmentation can provide (Bishop, 1995; Wu et al., 2020; Hansen and Wang, 2021), or by making it more difficult to overfit to spurious correlations (Raileanu et al., 2021; Shen et al., 2022). In this sense, we expect the insight of training with full data augmentation on a bigger subgroup $B \leq G$ from Theorem 1, to translate in practice to simply training on more diverse data, even data that is sampled from the same contexts.

5.2 Four Rooms

In this section, we demonstrate that increasing ensemble size and data diversity can significantly increase the generalisation performance of a distilled policy, even when most of the assumptions for Theorem 1 no longer hold. The Four Rooms grid world environment from the Minigrid benchmark

does not appear to have an invariant symmetry that plays a core part in generalising to new contexts, as required for the definition of a GTI-ZSPT. The teacher is an agent trained with Proximal Policy Optimisation (PPO Schulman et al., 2017) and is therefore not necessarily optimal in the training contexts. Additionally, the teacher is a stochastic policy that is distilled by regressing on the vector of probabilities or behaviour cloned using a logarithmic loss (see Appendix A.1 for more background on these losses).

5.2.1 Policy distillation improves generalisation

For the experiments in the Four Rooms environment the teacher is a policy trained with the PPO+Explore-Go algorithm for 8 million environment steps. The Explore-Go approach was introduced by Weltevrede et al. (2025) to increase generalisation by generating a more diverse training distribution for the RL agent. It leverages a separately trained pure exploration agent, rolled out at the beginning of each episode, to artificially increase the starting state distribution for the PPO agent.[2] Since this teacher trains on a more diverse state distribution than a normal PPO agent, it provides good teaching targets for our distillation datasets. We compare the following three datasets:

1. **Teacher:** This dataset consists of the teacher's trajectories in the (original) training contexts.
2. **Explore-Go:** This dataset mimics the training distribution for the Explore-Go approach by sampling teacher trajectories from additional starting states, generated by a pure exploration policy rolled out at the start of each episode. This dataset has the property that all the data is on-policy for our teacher, yet more diverse than the Teacher dataset.
3. **Mixed:** This dataset is a 50/50 mix of Teacher and trajectories collected by a separately trained pure exploration policy. This dataset is diverse, but does not solely consist of states encountered by the teacher.

In Table 3 we can see that the more diverse datasets (Mixed and Explore-Go) significantly outperform the Teacher dataset and that the ensemble of size $N = 10$ outperforms the single student $N = 1$ for each dataset type. Moreover, the ensemble, distilled on the Explore-Go dataset, generalises significantly better than the original PPO agent, whilst only requiring around 12% additional environment steps (compared to the teacher's training budget).

Table 3: Performance of an ensemble (of size N) of policy distillation or behaviour cloning policies on various datasets compared to the PPO+Explore-GO teacher in the Four Rooms environment. Shown are mean and standard deviation over 20 seeds, and in bold are the best returns including those with overlapping 95% confidence intervals (within the same category).

	Dataset	Train (N=1)	Train (N=10)	Test (N=1)	Test (N=10)
PPO+Explore-Go	-	**0.92 ± 0.020**	-	**0.74 ± 0.040**	-
Distillation	Teacher	**0.92 ± 0.020**	**0.92 ± 0.020**	0.56 ± 0.049	0.67 ± 0.054
	Mixed	**0.92 ± 0.020**	**0.92 ± 0.020**	0.72 ± 0.040	0.84 ± 0.034
	Explore-Go	**0.92 ± 0.020**	**0.92 ± 0.019**	0.78 ± 0.041	**0.88 ± 0.036**
Behaviour Cloning	Teacher	**0.91 ± 0.022**	**0.92 ± 0.020**	0.26 ± 0.046	0.37 ± 0.054
	Mixed	0.86 ± 0.031	**0.91 ± 0.025**	0.15 ± 0.024	0.20 ± 0.026
	Explore-Go	0.87 ± 0.028	**0.92 ± 0.021**	0.56 ± 0.060	**0.75 ± 0.045**

Lastly, we demonstrate in this section that the same insights also hold for a logarithmic behaviour cloning loss for stochastic policies that is widely used in practice (Foster et al., 2024). At the bottom of Table 3, we show that the an ensemble (of size $N = 10$), distilled on the Explore-Go dataset, generalises significantly better than a single behaviour cloning agent on the Teacher dataset. Note that behaviour cloning achieves lower performances than distillation, and that BC performs considerably worse on the Mixed dataset. In our definition of behaviour cloning, the student policy learns to imitate whatever policy collected the dataset, by only observing the actions that were actually sampled during collection. Therefore, the BC agent performs worse than the distillation agent, since the latter has access to more information (all the action probabilities of the teacher). On the Mixed dataset, the BC agent clones the behaviour policy that consists of a 50/50 mix of the (optimal) Teacher policy and (suboptimal) pure exploration policy. The resulting cloned behaviour performs even worse than

[2]For pure exploration, the objective focuses solely on exploring new parts of the state space, ignoring rewards.

the BC agent trained on the Teacher dataset. In contrast, the policy distillation agent on the Mixed dataset regresses on the action probabilities of the Teacher, on the states encountered by the 50/50 mixture of policies, and therefore has a much better learning target.

6 Discussion and limitations

The experiments in the Four Rooms environment in Section 5.2.1 serve to empirically demonstrate how our insights can be leveraged to significantly enhance the generalisation performance of a reinforcement learning agent through policy distillation. A clear example of this is seen in our ensemble $N = 10$, distilled on the Explore-Go dataset, which achieves substantially higher test performance than the original PPO+Explore-Go teacher policy (see Table 3). The potential of policy distillation after training as a tool to improve generalisation was initially identified in Lyle et al. (2022), but we believe the results of this paper provide a more compelling argument and empirical evidence for this phenomenon.

Whether the benefits of performing data augmentation with respect to some symmetry group actually stem from induced invariance or reduced overfitting and other forms of regularisation, is still an ongoing topic of discussion in the literature (Lyle et al., 2020; Shen et al., 2022). To add to this discussion, in Appendix D.1 we measure the invariance of our trained models on the 'Reacher with rotational symmetry' experiments from Table 1 and plot it against the ensemble size N and subgroup $B \leq SO(2)$. We find that in this particular experiment, the distilled policies *do* become more invariant as ensemble and subgroup size increase, just as our theory predicts.

Obtaining tight generalisation bounds for neural networks is notoriously challenging (Jiang et al., 2020; Gastpar et al., 2024). Moreover, some of the assumptions for Theorem 1, such as infinitely wide neural networks, are hard to meet in practice. Therefore, we believe the true strength of our theory lies in its ability to identify crucial properties of the dataset distribution and distilled ensemble that are capable of improving generalisation performance. Nonetheless, in Appendix D.2, we analyse how well our results fit the $a + \frac{b}{\sqrt{N}}$ relation identified by our theory. We find that our results reasonably agree with the shape of the theoretical upper bound, suggesting that our bound is not completely vacuous.

Finally, all ensemble members are trained independently, and during inference, can also be evaluated independently (an independent forward pass with an average over the output of the ensemble afterwards). This inherent independence means that both the training and inference processes of the ensemble are parallelisable. If parallelisation is not feasible, the runtime for both training and inference would increase linearly with the ensemble size. It is important to note that all ensemble members are distilled on the same dataset. This means the small number of additional environment steps required to sample this dataset is independent of ensemble size.

7 Conclusion

In this paper, we investigate the advantage of policy distillation for improving zero-shot policy transfer (ZSPT) in reinforcement learning. We introduce the generalisation through invariance ZSPT setting, to prove a generalisation bound for a policy distilled after training. Our analysis highlights two practical insights: to 1) distil an ensemble of policies, and to 2) distil it on a diverse set of states from the training contexts. We empirically evaluate that the insights hold in the Four Rooms environment from the Minigrid benchmark, even though it does not satisfy all the assumptions required for the theory, and that they also translate to the behaviour cloning setting. Moreover, we show that distilling an ensemble of policies on diverse set of states can produce a policy that generalises significantly better than the original RL agent, thus demonstrating that policy distillation can be a powerful tool to increase generalisation performance of reinforcement learning agents.

Acknowledgments and Disclosure of Funding

We thank Caroline Horsch, Laurens Engwegen and Oussama Azizi for fruitful discussions and feedback. The project has received funding from the EU Horizon 2020 programme under grant number 964505 (Epistemic AI) and was also partially funded by the Dutch Research Council (NWO) project *Reliable Out-of-Distribution Generalization in Deep Reinforcement Learning* with project number OCENW.M.21.234. The computational resources for empirical work were provided by the Delft AI Cluster (DAIC) (2024) and the Delft High Performance Computing Centre (DHPC) (2024).

References

S. Akshay, Nathalie Bertrand, Serge Haddad, and Loïc Hélouët. The Steady-State Control Problem for Markov Decision Processes. In Kaustubh R. Joshi, Markus Siegle, Mariëlle Stoelinga, and Pedro R. D'Argenio, editors, *Quantitative Evaluation of Systems - 10th International Conference, QEST 2013, Buenos Aires, Argentina, August 27-30, 2013. Proceedings*, volume 8054 of *Lecture Notes in Computer Science*, pages 290–304. Springer, 2013. doi: 10.1007/978-3-642-40196-1_26. URL https://doi.org/10.1007/978-3-642-40196-1_26.

Himani Arora, Rajath Kumar, Jason Krone, and Chong Li. Multi-task Learning for Continuous Control. *CoRR*, abs/1802.01034, 2018. URL http://arxiv.org/abs/1802.01034. arXiv: 1802.01034.

Aharon Azulay and Yair Weiss. Why do deep convolutional networks generalize so poorly to small image transformations? *J. Mach. Learn. Res.*, 20:184:1–184:25, 2019. URL https://jmlr.org/papers/v20/19-519.html.

Christopher M. Bishop. Training with Noise is Equivalent to Tikhonov Regularization. *Neural Comput.*, 7(1):108–116, 1995. doi: 10.1162/NECO.1995.7.1.108. URL https://doi.org/10.1162/neco.1995.7.1.108.

Maxime Chevalier-Boisvert, Bolun Dai, Mark Towers, Rodrigo de Lazcano, Lucas Willems, Salem Lahlou, Suman Pal, Pablo Samuel Castro, and Jordan Terry. Minigrid \& Miniworld: Modular \& Customizable Reinforcement Learning Environments for Goal-Oriented Tasks. *CoRR*, abs/2306.13831, 2023. URL https://minigrid.farama.org.

Karl Cobbe, Oleg Klimov, Christopher Hesse, Taehoon Kim, and John Schulman. Quantifying Generalization in Reinforcement Learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 1282–1289. PMLR, 2019. URL http://proceedings.mlr.press/v97/cobbe19a.html.

Wojciech M. Czarnecki, Razvan Pascanu, Simon Osindero, Siddhant M. Jayakumar, Grzegorz Swirszcz, and Max Jaderberg. Distilling Policy Distillation. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, volume 89 of *Proceedings of Machine Learning Research*, pages 1331–1340. PMLR, 2019. URL http://proceedings.mlr.press/v89/czarnecki19a.html.

Delft AI Cluster (DAIC). The Delft AI Cluster (DAIC), RRID:SCR_025091, 2024. URL https://doc.daic.tudelft.nl/.

Delft High Performance Computing Centre (DHPC). DelftBlue Supercomputer (Phase 2), 2024. URL https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase2.

Linxi Fan, Guanzhi Wang, De-An Huang, Zhiding Yu, Li Fei-Fei, Yuke Zhu, and Animashree Anandkumar. SECANT: Self-Expert Cloning for Zero-Shot Generalization of Visual Policies. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 3088–3099. PMLR, 2021. URL http://proceedings.mlr.press/v139/fan21c.html.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard H. Hovy. A Survey of Data Augmentation Approaches for NLP. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 968–988. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.FINDINGS-ACL.84. URL https://doi.org/10.18653/v1/2021.findings-acl.84.

Axel Flinth and Fredrik Ohlsson. Optimization Dynamics of Equivariant and Augmented Neural Networks. *CoRR*, abs/2303.13458, 2023. doi: 10.48550/ARXIV.2303.13458. URL https://doi.org/10.48550/arXiv.2303.13458. arXiv: 2303.13458.

Dylan J. Foster, Adam Block, and Dipendra Misra. Is Behavior Cloning All You Need? Understanding Horizon in Imitation Learning. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/da84e39ae51fd26bb5110d9659c06e13-Abstract-Conference.html.

Michael Gastpar, Ido Nachum, Jonathan Shafer, and Thomas Weinberger. Fantastic Generalization Measures are Nowhere to be Found. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=NkmJotfL42.

Jonas Geiping, Micah Goldblum, Gowthami Somepalli, Ravid Shwartz-Ziv, Tom Goldstein, and Andrew Gordon Wilson. How Much Data Are Augmentations Worth? An Investigation into Scaling Laws, Invariance, and Implicit Regularization. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=3aQs3MCSexD.

Jan E. Gerken and Pan Kessel. Emergent Equivariance in Deep Ensembles. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=plXXbXjvQ9.

Dibya Ghosh, Avi Singh, Aravind Rajeswaran, Vikash Kumar, and Sergey Levine. Divide-and-Conquer Reinforcement Learning. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=rJwelMbR-.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1856–1865. PMLR, 2018. URL http://proceedings.mlr.press/v80/haarnoja18b.html.

Assaf Hallak, Dotan Di Castro, and Shie Mannor. Contextual Markov Decision Processes. *CoRR*, abs/1502.02259, 2015. URL http://arxiv.org/abs/1502.02259. arXiv: 1502.02259.

Nicklas Hansen and Xiaolong Wang. Generalization in Reinforcement Learning by Soft Data Augmentation. In *IEEE International Conference on Robotics and Automation, ICRA 2021, Xi'an, China, May 30 - June 5, 2021*, pages 13611–13617. IEEE, 2021. doi: 10.1109/ICRA48506.2021.9561103.

Abdolhossein Hoorfar and Mehdi Hassani. Inequalities on the Lambert W function and hyperpower function. *J. Inequal. Pure and Appl. Math*, 9(2):5–9, 2008. URL https://scholar.googleusercontent.com/scholar.bib?q=info:ld5Z3TWCrwwJ:scholar.google.com/&output=citation&scisdr=ClHgGOi9EOmfyq37lfM:AFWwaeYAAAAAaBD9jfPyA3RJ6xRHSKpUCZOKjJA&scisig=AFWwaeYAAAAAaBD9jUtzEyvSjuBpylb1O4WCxnE&scisf=4&ct=citation&cd=-1&hl=en.

Maximilian Igl, Gregory Farquhar, Jelena Luketina, Wendelin Boehmer, and Shimon Whiteson. Transient Non-stationarity and Generalisation in Deep Reinforcement Learning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=Qun8fv4qSby.

Arthur Jacot, Clément Hongler, and Franck Gabriel. Neural Tangent Kernel: Convergence and Generalization in Neural Networks. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 8580–8589, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/5a4be1fa34e62bb8a6ec6b91d2462f5a-Abstract.html.

Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic Generalization Measures and Where to Find Them. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=SJgIPJBFvH.

Yiding Jiang, J. Zico Kolter, and Roberta Raileanu. On the Importance of Exploration for Generalization in Reinforcement Learning. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/2a4310c4fd24bd336aa2f64f93cb5d39-Abstract-Conference.html.

Sham M. Kakade and John Langford. Approximately Optimal Approximate Reinforcement Learning. In Claude Sammut and Achim G. Hoffmann, editors, *Machine Learning, Proceedings of the Nineteenth International Conference (ICML 2002), University of New South Wales, Sydney, Australia, July 8-12, 2002*, pages 267–274. Morgan Kaufmann, 2002.

Ken Kansky, Tom Silver, David A. Mély, Mohamed Eldawy, Miguel Lázaro-Gredilla, Xinghua Lou, Nimrod Dorfman, Szymon Sidor, D. Scott Phoenix, and Dileep George. Schema Networks: Zero-shot Transfer with a Generative Causal Model of Intuitive Physics. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1809–1818. PMLR, 2017. URL http://proceedings.mlr.press/v70/kansky17a.html.

Robert Kirk, Amy Zhang, Edward Grefenstette, and Tim Rocktäschel. A Survey of Zero-shot Generalisation in Deep Reinforcement Learning. *J. Artif. Intell. Res.*, 76:201–264, 2023. doi: 10.1613/JAIR.1.14174. URL https://doi.org/10.1613/jair.1.14174.

Henry Kvinge, Tegan Emerson, Grayson Jorgenson, Scott Vasquez, Tim Doster, and Jesse D. Lew. In What Ways Are Deep Neural Networks Invariant and How Should We Measure This? In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/d36dfcdb14473a8526111c221660f2ab-Abstract-Conference.html.

Jaehoon Lee, Lechao Xiao, Samuel S. Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide Neural Networks of Any Depth Evolve as Linear Models Under Gradient Descent. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8570–8581, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/0d1a9651497a38d8b1c3871c84528bd4-Abstract.html.

Kimin Lee, Kibok Lee, Jinwoo Shin, and Honglak Lee. Network Randomization: A Simple Technique for Generalization in Deep Reinforcement Learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=HJgcvJBFvB.

Chi-Heng Lin, Chiraag Kaushik, Eva L. Dyer, and Vidya Muthukumar. The good, the bad and the ugly sides of data augmentation: An implicit spectral regularization perspective. *CoRR*, abs/2210.05021, 2022. doi: 10.48550/ARXIV.2210.05021. URL https://doi.org/10.48550/arXiv.2210.05021. arXiv: 2210.05021.

Kaixiang Lin, Shu Wang, and Jiayu Zhou. Collaborative Deep Reinforcement Learning. *CoRR*, abs/1702.05796, 2017. URL http://arxiv.org/abs/1702.05796. arXiv: 1702.05796.

Clare Lyle, Mark van der Wilk, Marta Kwiatkowska, Yarin Gal, and Benjamin Bloem-Reddy. On the Benefits of Invariance in Neural Networks. *CoRR*, abs/2005.00178, 2020. URL https://arxiv.org/abs/2005.00178. arXiv: 2005.00178.

Clare Lyle, Mark Rowland, Will Dabney, Marta Kwiatkowska, and Yarin Gal. Learning Dynamics and Generalization in Deep Reinforcement Learning. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 14560–14581. PMLR, 2022. URL https://proceedings.mlr.press/v162/lyle22a.html.

Davide Maran, Alberto Maria Metelli, and Marcello Restelli. Tight Performance Guarantees of Imitator Policies with Continuous Actions. In Brian Williams, Yiling Chen, and Jennifer Neville, editors, *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pages 9073–9080. AAAI Press, 2023. doi: 10.1609/AAAI.V37I8.26089. URL https://doi.org/10.1609/aaai.v37i8.26089.

Ishita Mediratta, Qingfei You, Minqi Jiang, and Roberta Raileanu. The Generalization Gap in Offline Reinforcement Learning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=3w6xuXDOdY.

Ning Miao, Tom Rainforth, Emile Mathieu, Yann Dubois, Yee Whye Teh, Adam Foster, and Hyunjik Kim. Learning Instance-Specific Augmentations by Capturing Local Invariances. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 24720–24736. PMLR, 2023. URL https://proceedings.mlr.press/v202/miao23a.html.

Oskar Nordenfors and Axel Flinth. Ensembles provably learn equivariance through data augmentation. *CoRR*, abs/2410.01452, 2024. doi: 10.48550/ARXIV.2410.01452. URL https://doi.org/10.48550/arXiv.2410.01452. arXiv: 2410.01452.

Emilio Parisotto, Lei Jimmy Ba, and Ruslan Salakhutdinov. Actor-Mimic: Deep Multitask and Transfer Reinforcement Learning. In Yoshua Bengio and Yann LeCun, editors, *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/1511.06342.

Emmanuel Rachelson and Michail G. Lagoudakis. On the locality of action domination in sequential decision making. In *International Symposium on Artificial Intelligence and Mathematics, ISAIM 2010, Fort Lauderdale, Florida, USA, January 6-8, 2010*, 2010. URL http://gauss.ececs.uc.edu/Workshops/isaim2010/papers/lag.pdf.

Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-Baselines3: Reliable Reinforcement Learning Implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021. URL http://jmlr.org/papers/v22/20-1364.html.

Roberta Raileanu, Max Goldstein, Denis Yarats, Ilya Kostrikov, and Rob Fergus. Automatic Data Augmentation for Generalization in Reinforcement Learning. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 5402–5415, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/2b38c2df6a49b97f706ec9148ce48d86-Abstract.html.

Stéphane Ross, Geoffrey J. Gordon, and Drew Bagnell. A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. In Geoffrey J. Gordon, David B. Dunson, and Miroslav Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2011, Fort Lauderdale, USA, April 11-13, 2011*, volume 15 of *JMLR Proceedings*, pages 627–635. JMLR.org, 2011. URL http://proceedings.mlr.press/v15/ross11a/ross11a.pdf.

Andrei A. Rusu, Sergio Gomez Colmenarejo, Çaglar Gülçehre, Guillaume Desjardins, James Kirkpatrick, Razvan Pascanu, Volodymyr Mnih, Koray Kavukcuoglu, and Raia Hadsell. Policy Distillation. In Yoshua Bengio and Yann LeCun, editors, *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/1511.06295.

Fereshteh Sadeghi and Sergey Levine. CAD2RL: Real Single-Image Flight Without a Single Real Image. In Nancy M. Amato, Siddhartha S. Srinivasa, Nora Ayanian, and Scott Kuindersma, editors, *Robotics: Science and Systems XIII, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, July 12-16, 2017*, 2017. doi: 10.15607/RSS.2017.XIII.034. URL http://www.roboticsproceedings.org/rss13/p34.html.

Simon Schmitt, Jonathan J. Hudson, Augustin Zídek, Simon Osindero, Carl Doersch, Wojciech M. Czarnecki, Joel Z. Leibo, Heinrich Küttler, Andrew Zisserman, Karen Simonyan, and S. M. Ali Eslami. Kickstarting Deep Reinforcement Learning. *CoRR*, abs/1803.03835, 2018. URL http://arxiv.org/abs/1803.03835. arXiv: 1803.03835.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms. *CoRR*, abs/1707.06347, 2017. URL http://arxiv.org/abs/1707.06347. arXiv: 1707.06347.

Ruoqi Shen, Sébastien Bubeck, and Suriya Gunasekar. Data Augmentation as Feature Manipulation. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 19773–19808. PMLR, 2022. URL https://proceedings.mlr.press/v162/shen22a.html.

Connor Shorten and Taghi M. Khoshgoftaar. A survey on Image Data Augmentation for Deep Learning. *J. Big Data*, 6:60, 2019. doi: 10.1186/S40537-019-0197-0. URL https://doi.org/10.1186/s40537-019-0197-0.

Miguel Suau, Matthijs T. J. Spaan, and Frans A. Oliehoek. Bad Habits: Policy Confounding and Out-of-Trajectory Generalization in RL. *Reinforcement Learning Journal*, 4:1711–1732, 2024. URL https://rlj.cs.umass.edu/2024/papers/Paper216.html.

Yee Whye Teh, Victor Bapst, Wojciech M. Czarnecki, John Quan, James Kirkpatrick, Raia Hadsell, Nicolas Heess, and Razvan Pascanu. Distral: Robust multitask reinforcement learning. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4496–4506, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/0abdc563a06105aee3c6136871c9f4d1-Abstract.html.

Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop, ITW 2015, Jerusalem, Israel, April 26 - May 1, 2015*, pages 1–5. IEEE, 2015. doi: 10.1109/ITW.2015.7133169.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2017, Vancouver, BC, Canada, September 24-28, 2017*, pages 23–30. IEEE, 2017. doi: 10.1109/IROS.2017.8202133.

Aaron Walsman, Muru Zhang, Sanjiban Choudhury, Dieter Fox, and Ali Farhadi. Impossibly Good Experts and How to Follow Them. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=sciA_xgYofB.

Xudong Wang, Long Lian, and Stella X. Yu. Unsupervised Visual Attention and Invariance for Reinforcement Learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 6677–6687. Computer Vision Foundation / IEEE, 2021. doi: 10.1109/CVPR46437.2021.00661. URL https://openaccess.thecvf.com/content/CVPR2021/html/Wang_Unsupervised_Visual_Attention_and_Invariance_for_Reinforcement_Learning_CVPR_2021_paper.html.

Max Weltevrede, Caroline Horsch, Matthijs T. J. Spaan, and Wendelin Böhmer. Exploration Implies Data Augmentation: Reachability and Generalisation in Contextual MDPs, March 2025. URL http://arxiv.org/abs/2410.03565. arXiv:2410.03565 [cs].

Sen Wu, Hongyang R. Zhang, Gregory Valiant, and Christopher Ré. On the Generalization Effects of Linear Transformations in Data Augmentation. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 10410–10420. PMLR, 2020. URL http://proceedings.mlr.press/v119/wu20g.html.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Commun. ACM*, 64(3):107–115, 2021. doi: 10.1145/3446776.

A Extended background

A.1 Policy distillation & behaviour cloning

In policy distillation, a knowledge transfer occurs by distilling a policy from a *teacher* network into a newly initialised *student* network. Depending on the objective of the knowledge transfer, the student network can be smaller, the same size, or bigger than the teacher network. Moreover, there are many different ways the policy can be distilled (Czarnecki et al., 2019), depending on the specific loss function used (Ghosh et al., 2018; Teh et al., 2017), whether the student can collect additional data during distillation (Lin et al., 2017; Parisotto et al., 2016; Ross et al., 2011), or has access to additional information like rewards or a teacher's value function (Czarnecki et al., 2019).

We consider a student network with the same architecture and size as the teacher that is distilled on a fixed dataset (so without allowing additional interactions of the student with the environment). This fixed dataset is collected after training, and usually consists of on-policy data collected by the teacher itself. In this paper, we analyse a simplified setting where both the student and teacher policy are assumed to be deterministic and scalar: $\pi_\theta : S \to \mathbb{R}, \pi_\beta : S \to \mathbb{R}$. A simple distillation loss in this setting is the mean squared error (MSE) between the output of the two policies:

$$l_D(\theta, \mathcal{D}, \pi_\beta) = \frac{1}{n} \sum_{s \in \mathcal{D}} (\pi_\theta(s) - \pi_\beta(s))^2$$

where $\mathcal{D} = \{s_1, ..., s_n\}$ is the set of states we distil on.

More generally, distillation can be performed between deterministic, vector valued student and teacher policies $\pi_\theta : S \to \mathbb{R}^d, \pi_\beta : S \to \mathbb{R}^d$ with the loss

$$l(\theta, \mathcal{D}, \pi_\beta) = \frac{1}{n} \sum_{s \in \mathcal{D}} ||\pi_\theta(s) - \pi_\beta(s)||_2^2 \tag{6}$$

For stochastic policies, it is more common to minimise the Kullback-Leibler (KL) divergence between the student and teacher policies (Arora et al., 2018), sometimes including an entropy regularisation term (Teh et al., 2017; Lyle et al., 2022)

$$l(\theta, \mathcal{D}, \pi_\beta) = \frac{1}{n} \sum_{s \in \mathcal{D}} D_{KL}(\pi_\theta(s)||\pi_\beta(s)) + \lambda H(\pi_\theta)$$

where $H(\cdot)$ denotes the entropy of the policy. An alternative approach for discrete, stochastic policies is to regress towards the logits or probabilities over actions from the teacher:

$$l(\theta, \mathcal{D}, \pi_\beta) = \frac{1}{n} \sum_{s \in \mathcal{D}} ||\pi_\theta(\cdot|s) - \pi_\beta(\cdot|s)||_2^2 \tag{7}$$

where $\pi(\cdot|s)$ now indicates the vector (of dimension $|A|$) of probabilities or logits that policy π produces in state s.

As mentioned above, usually the policy is distilled on on-policy data collected by the teacher. However, in general, a policy can in principle be distilled on any distribution over states, since the targets produced by the teacher (i.e., $\pi_\beta(s)$ or $\pi_\beta(\cdot|s)$) can be trained off-policy, independently of how the state s was reached, or which action was taken in s during collection.

A.1.1 Behaviour cloning

We consider *behaviour cloning* (BC) as a specific instance of policy distillation, where the student network only has access to a fixed dataset of the teacher's behaviour (state-action tuples) and not additional information like the teacher's policy, value function or environment rewards. The goal in behaviour cloning is to learn to imitate the behaviour policy (e.g., the teacher) that collected the dataset. In this sense, it differs from the general distillation setting, in that the learning targets are always on-policy with respect to the policy (or the mixture of policies) that collected the data.

Just as for distillation, the BC loss can differ depending on whether the student policy is deterministic or stochastic. For deterministic policies, the loss is usually the MSE between the student and the action observed in the dataset:

$$l(\theta, \mathcal{D}_\beta) = \frac{1}{n} \sum_{i=0}^{n} ||\pi_\theta(s_i) - \vec{a_i}||_2^2$$

where $\mathcal{D}_\beta = \{(s_1, a_1), ..., (s_n, a_n)\}$ is a dataset of behaviour of size n. For stochastic policies, it is more common to use a logarithmic loss (Foster et al., 2024)

$$l_(\theta, \mathcal{D}_\beta) = -\sum_{i=0}^{n} \ln \pi_\theta(a_i|s_i) \tag{8}$$

Note that these losses mainly differ from the distillation losses in that the learning targets are the actions a_i taken by the policy that collected the dataset, rather than the (potentially off-policy) teacher policy $\pi_\beta(s_i)$.

A.2 Group Symmetry

A group is a non-empty set G together with a binary operation $\cdot$ that satisfies the following requirements:

$$\begin{aligned}
a \cdot b \in G, \quad \forall a, b \in G \qquad & \text{(Closure)} \\
(a \cdot b) \cdot c = a \cdot (b \cdot c), \quad \forall a, b, c \in G \qquad & \text{(Associativity)} \\
\exists e \in G, \quad e \cdot a = a \cdot e = a, \quad \forall a \in G \qquad & \text{(Identity)} \\
\forall a \in G, \exists a^{-1} \in G, \quad a \cdot a^{-1} = a^{-1} \cdot a = e \qquad & \text{(Inverse)}
\end{aligned}$$

We will abuse notation slightly by denoting both the group and the non-empty set with G, depending on context.

We can define a *group representation* ψ_X acting on X, as a map $\psi : G \to \mathrm{GL}(X)$ from G to the general linear group $\mathrm{GL}(X)$ of a vector space X, where the general linear group is defined as the set of $n \times n$ invertible matrices (for finite dimensional vector space X with dimension n) with matrix multiplication as operator and where the map ψ is a group homomorphism, i.e. $\psi(a)\psi(b) = \psi(a \cdot b), \quad \forall a, b \in G$. With these definitions, invariance of a function f is defined as

$$f(\psi_X(g)x) = f(x) \quad \forall x \in X, g \in G$$

A useful property when performing full data augmentation with a group G, is that applying a transformation from G to any of the training samples in $\mathcal{T}_G$, is equivalent to applying a permutation $\mathtt{p}_g$ the augmented training dataset indices:

$$\psi_X(g)x_i = x_{\mathtt{p}_g(i)}, \qquad \text{where } i \in \{1, ..., |\mathcal{T}_G|\} \tag{9}$$

A.3 The infinite width limit

In the limit of infinite layer width, an ensemble of neural networks from random initialization follows a Gaussian process that is characterised by the neural tangent kernel (NTK Jacot et al., 2018) defined as

$$\Theta(x, x') = \sum_{l=1}^{L} \mathbb{E}_{\theta\sim\mu}\left[\left(\frac{\partial f_\theta(x)}{\partial \theta^{(l)}}\right)^T \left(\frac{\partial f_\theta(x')}{\partial \theta^{(l)}}\right)\right],$$

where we assumed the network f_θ has L layers and $\theta^{(l)}$ denotes the parameters at layers $l \in [1, L]$ respectively. The Gaussian process at time t has mean m_t and covariance Σ_t (Lee et al., 2019):

$$\begin{aligned}
m_t(x) &= \Theta(x, x_i)[\Theta^{-1}T_t]_{ij}y_j \\
\Sigma_t(x, x') &= \mathcal{K}(x, x') + \Sigma_t^{(1)}(x, x') - (\Sigma_t^{(2)}(x, x') + \text{ h.c.})
\end{aligned}$$

where we use the Einstein notation convention to indicate implicit sums over the dataset indices i, j, h.c. indicates the Hermitian conjugate of the preceding term, $T_t = (\mathbb{I} - \exp(-\eta\Theta t))$, $\mathcal{K}(x, x') = \mathbb{E}_{\theta\sim\mu}[f_\theta(x)f_\theta(x')]$ is the neural network Gaussian process (NNGP) kernel, and $\Sigma_t^{(1)}$ and $\Sigma_t^{(2)}$ are defined as follows:

$$\begin{aligned}
\Sigma_t^{(1)}(x, x') &= \Theta(x, x_i)[\Theta^{-1}T_t\mathcal{K}T_t\Theta^{-1}]_{ij}\Theta(x_j, x') \\
\Sigma_t^{(2)}(x, x') &= \Theta(x, x_i)[\Theta^{-1}T_t]_{ij}\,\mathcal{K}(x_j, x').
\end{aligned}$$

We use shorthand notation $\Sigma_t(x, x) = \Sigma_t(x)$ for the NNGP variance.

An infinite ensemble $\bar{f}_t$ equals the mean m_t of the Gaussian process: $\bar{f}_t(x) = m_t(x)$. Note that for $t \to \infty$, the output of the infinite ensemble $\bar{f}_\infty$ on the training inputs $\mathcal{X}$ converges to the targets $\mathcal{Y}$:

$$\bar{f}_\infty(\mathcal{X}) = m_\infty(\mathcal{X}) = \Theta(\mathcal{X}, \mathcal{X})\Theta(\mathcal{X}, \mathcal{X})^{-1} T_\infty \mathcal{Y} = \mathcal{Y}$$

B Proof of Theorem 1

In this section, we will go through the steps for the proof of the main theorem of section 4. We first repeat the definition of the GTI-ZSPT and associated discrepency measure κ

Definition 1 (Generalisation through invariance ZSPT)**.** *Let $\mathcal{M}|_C$ be a CMDP and let $C_{train}, C_{test} \subset C$ be a set of training and testing contexts that define a ZSPT problem. Additionally, let π^* be the optimal policy in $\mathcal{M}|_C$, $S^{\pi^*}_{\mathcal{M}|_C} = \{s \in S | \rho^{\pi^*}_{\mathcal{M}|_C}(s) > 0\}$ denote the set of states with non-zero support under the on-policy distribution $\rho^{\pi^*}_{\mathcal{M}|_C}$ in CMDP $\mathcal{M}|_C$. In the generalisation through invariance ZSPT (GTI-ZSPT), the sets $S^{\pi^*}_{\mathcal{M}|_C}$ and $S^{\pi^*}_{\mathcal{M}|_{C_{train}}}$ admit a symmetric structure:*

$$S^{\pi^*}_{\mathcal{M}|_C} = \{\psi_S(g)s | g \in G, s \in \bar{S}\}$$
$$S^{\pi^*}_{\mathcal{M}|_{C_{train}}} = \{\psi_S(b)s | b \in B, s \in \bar{S}\}, \quad B \leq G$$

where $\bar{S} \subset S^{\pi^*}_{\mathcal{M}|_{C_{train}}}$ is a proper subset of $S^{\pi^*}_{\mathcal{M}|_{C_{train}}}$ and G is a non-trivial symmetry group (and $B \leq G$ a finite subgroup) that leaves the optimal policy invariant: $\pi^*(s) = \pi^*(\psi_S(g)s), \forall s \in \bar{S}$.

Definition 2. *For the group G and its finite subgroup $B \leq G$ that define the symmetric structure of a GTI-ZSPT (Definition 1), κ is a measure of discrepancy between the representations of these groups:*

$$\kappa = \max_{g \in G} \min_{b \in B} ||\psi_S(g) - \psi_S(b)||_{op}$$

where $||\cdot||_{op}$ denotes the operator norm.

B.1 Invariance of an ensemble

In order to prove Theorem 1, we first repeat Lemma 6.2 from Gerken and Kessel (2024) that bounds the invariance of an infinitely large ensemble of infinitely wide neural networks trained with full data augmentation on some finite subgroup $B \leq G$:

Lemma 6.2. *Let $\pi_\theta : S \to \mathbb{R}$ be an infinitely wide neural network with parameters θ and with Lipschitz continuous derivatives with respect to the parameters. Furthermore, let $\bar{\pi}_t$ be an infinite ensemble $\bar{\pi}_t(s) = \mathbb{E}_{\theta \sim \mu}[\pi_{\mathcal{L}_t \theta}(s)]$, where the initial weights θ are sampled from a distribution μ and the operator $\mathcal{L}_t$ maps θ to its corresponding value after t steps of gradient descent with respect to a MSE loss function. Define the error κ as a measure of discrepancy between representations from the group G and its finite subgroup B:*

$$\kappa = \max_{g \in G} \min_{b \in B} ||\psi_S(g) - \psi_S(b)||_{op} \tag{10}$$

The prediction of an infinite ensemble trained with full data augmentation on $B \leq G$ deviates from invariance by

$$\left|\bar{\pi}_t(s) - \bar{\pi}_t(\psi_S(g)s)\right| \leq \kappa\, C_\Theta(s), \qquad \forall g \in G \tag{11}$$

for any time t. Here $s \in S$ can by any state and C_Θ is independent of g.

Proof. For completeness we repeat the proof in our own notation in E.2. □

Next, we prove a lemma that bounds the prediction error between a finite ensemble and an infinite ensemble:

Lemma 1. *The difference between the infinite ensemble $\bar{\pi}_t$ and its finite Monte Carlo estimate $\hat{\pi}_t$ of size N, is bounded by*

$$|\bar{\pi}_t(s) - \hat{\pi}_t(s)| \leq \frac{1}{\sqrt{N}} C_{\Sigma_t}(s, \epsilon) \tag{12}$$

with probability at least $1 - \epsilon$. Here Σ_t is the variance of the NNGP at time t and $C_{\Sigma_t}(s, \epsilon)$ depends on Σ_t, the state s and confidence level ϵ.

Proof. We start with Lemma B.4 from Gerken and Kessel (2024) that holds for any Monte-Carlo estimator:

Lemma B.4. *The probability that the deep ensemble $\bar{\pi}_t$ and its Monte-Carlo estimate $\hat{\pi}_t$ differ by more than a given threshold δ is bounded by*

$$\mathbb{P}\big[|\bar{\pi}_t(s) - \hat{\pi}_t(s)| > \delta\big] \leq \sqrt{\frac{2}{\pi}}\frac{\sigma_s}{\delta}\exp\Big(-\frac{\delta^2}{2\sigma_s^2}\Big),$$

where we have defined

$$\sigma_s^2 := Var(\hat{\pi}_t)(s) = \frac{\Sigma_t(s)}{N}$$

where $\Sigma_t(s)$ is the NNGP variance and N is the finite ensemble size.

Proof. For completeness we repeat the proof in our own notation in E.3. □

We can use this lemma to bound the probability of the deviation between finite and infinite ensemble to be smaller than a threshold δ:

$$\begin{aligned}\mathbb{P}\big[|\bar{\pi}_t(s) - \hat{\pi}_t(s)| \leq \delta\big] &= 1 - \mathbb{P}\big[|\bar{\pi}_t(s) - \hat{\pi}_t(s)| > \delta\big] \\ &\geq 1 - \epsilon\end{aligned}$$

where $\epsilon = \sqrt{\frac{2}{\pi}}\frac{\sigma_s}{\delta}\exp\big(-\frac{\delta^2}{2\sigma_s^2}\big)$. Next, we rewrite δ in terms of a given confidence level ϵ:

$$\begin{aligned}\epsilon &= \sqrt{\frac{2}{\pi}}\frac{\sigma_s}{\delta}\exp\big(-\frac{\delta^2}{2\sigma_s^2}\big) \\ \frac{2}{\pi\epsilon^2} &= \frac{\delta^2}{\sigma_s^2}\exp\big(\frac{\delta^2}{\sigma_s^2}\big) \\ \frac{\delta^2}{\sigma_s^2} &= W_0\big(\frac{2}{\pi\epsilon^2}\big) \\ \delta &= \sigma_s\sqrt{W_0\big(\frac{2}{\pi\epsilon^2}\big)}\end{aligned}$$

where W_0 is the principal branch of the Lambert W function and the second to last step holds because $\frac{\delta^2}{\sigma_s^2}, \frac{2}{\pi\epsilon^2} \in \mathbb{R}$ and $\frac{2}{\pi\epsilon^2} \geq 0$ for a given probability ϵ. If we know the value for ϵ, $W_0(\frac{2}{\pi\epsilon^2})$ can be solved for numerically. However, in general, the principal branch of the Lambert W function has no closed-form solution, but was upper bounded by Hoorfar and Hassani (2008)

$$W_0(x) \leq \ln\left(\frac{2x+1}{1+\ln(x+1)}\right)$$

for $x \geq -1/e$. Which means we can upper bound δ with:

$$\delta \leq \sqrt{\frac{\Sigma_t(s)}{N}}\sqrt{\ln\left(\frac{4+\pi\epsilon^2}{\pi\epsilon^2 + \pi\epsilon^2\ln(2+\pi\epsilon^2) + \pi\epsilon^2\ln(\pi\epsilon^2)}\right)} \tag{13}$$

$$\leq \frac{1}{\sqrt{N}}C_{\Sigma_t}(s,\epsilon) \tag{14}$$

□

We can now prove an intermediate lemma that bounds the deviation from the optimal policy for a finite ensemble (rather than an infinite one, as in Lemma 6.2)

Lemma 2. *Let the student policy $\hat{\pi}_\infty$ be an ensemble of N infinitely wide neural networks π_θ with Lipschitz continuous derivatives with respect to its parameters, distilled on an on-policy dataset $\mathcal{D} = S^{\pi_\beta}_{\mathcal{M}|_{C_{train}}}$ consisting of all the states encountered by the teacher in $\mathcal{M}|_{C_{train}}$.*

If the teacher is optimal in the training tasks C_{train} (but arbitrarily bad anywhere else), the deviation from the optimal policy for any test state $s' \in S^{\pi^*}_{\mathcal{M}|_{C_{test}}}$ is bounded with probability at least $1-\epsilon$, by:

$$|\pi^*(s') - \hat{\pi}_\infty(s')| \leq \kappa C_\Theta(s') + \frac{1}{\sqrt{N}}C_{\Sigma_\infty}(s',\epsilon)$$

where κ is the measure of discrepancy between subgroup $B \leq G$ and full group G (see definition 2) and $C_\Theta, C_{\Sigma_\infty}$ depend on the state $s' \in S^{\pi^*}_{\mathcal{M}|_{C_{test}}}$, the NTK Θ (i.e. network architecture), and the dataset $\mathcal{D}$. Additionally, C_{Σ_∞} also depends on the NNGP kernel $\mathcal{K}$ (i.e. network initialisation) and the confidence level ϵ.

Proof. Because we assume the teacher is optimal in the training tasks, our training dataset is actually $\mathcal{D} = S^{\pi_\beta}_{\mathcal{M}|_{C_{train}}} = S^{\pi^*}_{\mathcal{M}|_{C_{train}}}$. Furthermore, by definition of the GTI-ZSPT setting, we have for the states encountered by the optimal policy in $\mathcal{M}|_C$: $S^{\pi^*}_{\mathcal{M}|_C} = \{\psi_S(g)s | g \in G, s \in \bar{S}\}$. Furthermore, we have for the states encountered by the optimal policy in $\mathcal{M}|_{C_{train}}$: $\mathcal{M}|_C$: $S^{\pi^*}_{\mathcal{M}|_{C_{train}}} = \{\psi_S(b)s | b \in B, s \in \bar{S}\}$ for $B \leq G$. This means that for any state $s \in S^{\pi^*}_{\mathcal{M}|_C}$, there exists a symmetry transformation $g^{-1} \in G$ from s to a state $\bar{s} \in \bar{S} \subset \mathcal{D}$ in the training dataset that leaves the policy invariant:

$$\forall s \in S^{\pi^*}_{\mathcal{M}|_C}, \quad \exists g^{-1} \in G, \qquad s.t. \qquad \psi_S(g^{-1})s = \bar{s} \wedge \pi^*(s) = \pi^*(\bar{s}), \quad \text{for some } \bar{s} \in \mathcal{D} \tag{15}$$

Since this holds for any state in $S^{\pi^*}_{\mathcal{M}|_C}$, it also holds for any state in $S^{\pi^*}_{\mathcal{M}|_{C_{test}}} \subset S^{\pi^*}_{\mathcal{M}|_C}$.

Now, Lemma 6.2 holds for any state $s \in S$ and any $g \in G$. So, if we choose $s = \bar{s}$ and g such that $\psi_S(g)\bar{s} = s'$ for a testing state $s' \in S^{\pi^*}_{\mathcal{M}|_{C_{test}}}$, we have

$$\begin{aligned} \left|\bar{\pi}_t(\bar{s}) - \bar{\pi}_t(\psi_S(g)\bar{s})\right| &\leq \kappa\, C_\Theta(\bar{s}) \\ \left|\bar{\pi}_t(\bar{s}) - \bar{\pi}_t(s')\right| &\leq \kappa\, C_\Theta(s') \end{aligned}$$

where we write that C_Θ is now a function of $s' \in S^{\pi^*}_{\mathcal{M}|_{C_{test}}}$ instead of $\bar{s} \in \mathcal{D}$, which we can do because there exists a one-to-one mapping between the two: $\bar{s} = \psi_S(g^{-1})s'$. The above bound holds for any time t. If we choose $t \to \infty$, we have that the infinite ensemble of infinitely wide neural networks $\bar{\pi}_t$ trained on $\mathcal{D}$, will converge to $\bar{\pi}_\infty(\bar{s}) = \pi_\beta(\bar{s}) = \pi^*(\bar{s})$, $\forall \bar{s} \in \mathcal{D}$. Furthermore, due to our choice of $g \in G$, we have that $\bar{\pi}_\infty(\bar{s}) = \pi^*(\bar{s}) = \pi^*(s')$, and the bound becomes

$$\left|\pi^*(s') - \bar{\pi}_\infty(s')\right| \leq \kappa\, C_\Theta(s'), \qquad \forall s' \in S^{\pi^*}_{\mathcal{M}|_{C_{test}}}$$

This bounds the output of the infinite ensemble after training $\bar{\pi}_\infty$, evaluated in a testing state $s' \in S^{\pi^*}_{\mathcal{M}|_{C_{test}}}$, to the optimal policy in that state. We can now combine this bound with our Lemma 1 above to bind the policy of a finite ensemble to the optimal policy in any testing state:

$$\begin{aligned} |\pi^*(s') - \hat{\pi}_\infty(s')| &= |\pi^*(s') - \bar{\pi}_\infty(s') + \bar{\pi}_\infty(s') - \hat{\pi}_\infty(s')| \\ &\leq |\pi^*(s') - \bar{\pi}_\infty(s')| + |\bar{\pi}_\infty(s') - \hat{\pi}_\infty(s')| \\ &\leq \kappa\, C_\Theta(s') + |\bar{\pi}_\infty(s') - \hat{\pi}_\infty(s')| \\ &\leq \kappa\, C_\Theta(s') + \frac{1}{\sqrt{N}} C_{\Sigma_\infty}(s', \epsilon) \quad \text{with probability } \geq 1 - \epsilon, \quad \forall s' \in S^{\pi^*}_{\mathcal{M}|_{C_{test}}} \end{aligned}$$

□

B.2 Performance during testing

We can now use Theorem 3 from Maran et al. (2023) to prove a performance bound for our student policy $\hat{\pi}_\infty(s')$ in the testing CMDP $\mathcal{M}|_{C_{test}}$ in terms of the Wasserstein distance between the student and optimal policy in this testing CMDP:

Theorem 3. *Let π^* be the optimal policy and $\hat{\pi}_\infty$ be the student policy. If the CMDP is (L_T, L_R)-Lipschitz continuous and the optimal and student policies are L_π-Lipschitz continuous, and we have that $\gamma L_T(1 + L_{\hat{\pi}_\infty}) < 1$, then it holds that:*

$$J^{\pi^*} - J^{\hat{\pi}_\infty} \leq \frac{L_R}{(1-\gamma)(1 - \gamma L_T(1 + L_{\hat{\pi}_\infty}))} \mathbb{E}_{s \sim d^{\pi^*}}[\mathcal{W}(\pi^*(\cdot|s), \hat{\pi}_\infty(\cdot|s))]$$

where $d^{\pi^*}(s) = (1-\gamma)\sum_{t=0}^{\infty} \gamma^t \mathbb{P}(s_t = s | \pi^*, p_0)$ is the γ-discounted visitation distribution and γ the the discount factor.

Proof. For completeness we repeat the proof in our notation in Appendix E.1. □

With this, we can finally prove the main theorem:

Theorem 1. *Consider policy distillation for a deterministic, scalar teacher policy $\pi_\beta : S \to \mathbb{R}$ (Equation (1) in Section 2.1) in a L_T, L_R-Lipschitz continuous CMDP in the GTI-ZSPT setting. Let the student policy $\hat{\pi}_\infty$ be an ensemble of N infinitely wide neural networks $\pi_\theta : S \to \mathbb{R}$ with Lipschitz continuous derivatives with respect to its parameters, distilled on an on-policy dataset $\mathcal{D} = S^{\pi_\beta}_{\mathcal{M}|_{C_{train}}} = \{\psi_S(b)s | b \in B, s \in \bar{S}\}$ consisting of all the states in the training contexts encountered by the teacher in the GTI-ZSPT setting. Furthermore, let the student policy be $L_{\hat{\pi}_\infty}$-Lipschitz continuous and assume $\gamma L_T(1 + L_{\hat{\pi}_\infty}) < 1$.*

If the teacher is optimal in the training tasks C_{train} (but arbitrarily bad anywhere else), the performance of the student in the testing CMDP $\mathcal{M}|_{C_{test}}$ is bounded with probability at least $1 - \epsilon$, by:

$$J^{\pi^*} - J^{\hat{\pi}_\infty} \leq \frac{L_R}{(1-\gamma)(1-\gamma L_T(1+L_{\hat{\pi}_\infty}))} \left(\kappa \bar{C}_\Theta + \frac{1}{\sqrt{N}} \bar{C}_{\Sigma_\infty}(\epsilon) \right) \tag{5}$$

where κ is the measure of discrepancy between subgroup $B \leq G$ and full group G (see definition 2) and $\bar{C}_\Theta, \bar{C}_{\Sigma_\infty}$ are constants that depend on the γ-discounted visitation distribution of the optimal policy in $\mathcal{M}|_{C_{test}}$, the network architecture, and the dataset $\mathcal{D}$. Additionally, $\bar{C}_{\Sigma_\infty}$ also depends on the network initialisation and the confidence level ϵ.

Proof. We have for deterministic polices that the Wasserstein distance reduces to

$$\mathcal{W}(\pi^*(\cdot|s), \hat{\pi}_\infty(\cdot|s)) = |\pi^*(s) - \hat{\pi}_\infty(s)|$$

So, if we invoke Theorem 3 from Maran et al. (2023) on the testing CMDP $\mathcal{M}|_{C_{test}}$, we can use Lemma 2 to show

$$\begin{aligned} J^{\pi^*} - J^{\hat{\pi}_\infty} &\leq \frac{L_R}{(1-\gamma)(1-\gamma L_T(1+L_{\hat{\pi}_\infty}))} \mathbb{E}_{s\sim d^{\pi^*}}[\mathcal{W}(\pi^*(\cdot|s), \hat{\pi}_\infty(\cdot|s))] \\ &\leq \frac{L_R}{(1-\gamma)(1-\gamma L_T(1+L_{\hat{\pi}_\infty}))} \mathbb{E}_{s\sim d^{\pi^*}}[|\pi^*(s) - \hat{\pi}_\infty(s)|] \\ &\leq \frac{L_R}{(1-\gamma)(1-\gamma L_T(1+L_{\hat{\pi}_\infty}))} \left(\kappa \bar{C}_\Theta + \frac{1}{\sqrt{N}} \bar{C}_{\Sigma_\infty}(\epsilon) \right) \end{aligned}$$

where $\bar{C}_\Theta = \mathbb{E}_{s\sim d^{\pi^*}}[C_\Theta(s)]$ depends on the NTK Θ (i.e. network architecture) and $C_{\Sigma_\infty}(\epsilon) = \mathbb{E}_{s\sim d^{\pi^*}}[C_{\Sigma_\infty}(s, \epsilon)]$ depends on the NNGP kernel $\mathcal{K}$ (i.e. network initialisation). □

C Experimental details

The code for all the experiments in the main text can be found at https://github.com/MWeltevrede/distillation-after-training.

C.1 'Reacher with rotational symmetry' CMDP

In the 'Reacher with rotational symmetry' CMDP from Figure 1, the state $s = (x_s, y_s, x_e, y_e, x_h, y_h)$ consists of the 2D Euclidean coordinates of the shoulder (x_s, y_s), elbow (x_e, y_e) and hand (x_h, y_h) centred around the target location, and the continuous 2D action space consists of the torque to rotate the shoulder and elbow joints. The episode terminates and the agent receives a reward of 1 if the hand of the robot arm is within a small area around the target location. Elsewhere, the reward function equals $\frac{1-0.5d_{target}}{0.5T}\delta_{d_{target}<d_{min}}$, where d_{target} is the distance between the target location and hand, $T = 200$ is the maximum number of steps before timeout, and $\delta_{d_{target}<d_{min}}$ is 1 only when the current d_{target} is smaller than the minimal distance d_{min} to target achieved in that episode, and 0 otherwise. In the experiments from Section 5.1, policies are distilled on datasets collected by rolling out trajectories from a teacher agent in a fixed set of training contexts. Ensembles are created by independently distilling N policies (with different seeds) and afterwards evaluating by averaging over the output of the N polices.

C.1.1 Satisfying the assumptions for the GTI-ZSPT setting

The 'Reacher with rotational symmetry' CMDP from Figure 1 satisfies the symmetric structure assumed in the GTI-ZSPT setting. To illustrate this, we could define the states encountered by the optimal policy in context 1 in Figure 1 as the subset of states $\bar{S}$ in the GTI-ZSPT definition. With that definition we can see that subgroup $B = C_4$ would generate all the states in $S^{\pi^*}_{\mathcal{M}|_{C_{train}}}$, and full group $G = SO(2)$ would generate all the states in $S^{\pi^*}_{\mathcal{M}|_C}$.

For the states $s = (x_s, y_s, x_e, y_e, x_h, y_h)$, the representation $\psi_S(\alpha)$ for a rotation with angle α is the block diagonal matrix:

$$\psi_S(\alpha) = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 & 0 & 0 & 0 \\ \sin\alpha & \cos\alpha & 0 & 0 & 0 & 0 \\ 0 & 0 & \cos\alpha & -\sin\alpha & 0 & 0 \\ 0 & 0 & \sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & 0 & 0 & \cos\alpha & -\sin\alpha \\ 0 & 0 & 0 & 0 & \sin\alpha & \cos\alpha \end{bmatrix}$$

which is orthogonal.

Additionally, the CMDP is L_T-Lipschitz continuous since there are no collisions causing non-smooth transitions. Moreover, with a proper choice of reward function (for example, $R = \frac{1}{d_{target}}$), it is also L_R-Lipschitz continuous. Note that for our experiments, we choose a non smooth reward function since it helped with training the teacher.

C.1.2 Figure 1 & Table 1

In the setting from Figure 1 and Table 1, the contexts only differ in the location of the shoulder. The robot arm pose always starts at a $45°$ degree angle for the shoulder joint (counter-clockwise with respect to an axis drawn from the shoulder to the target), and a $90°$ degree angle for the elbow joint (clockwise with respect to an axis drawn from the shoulder to elbow). In the testing distribution, the shoulder can be located anywhere along a circle around the target location, but the starting pose is always the same.

For the training contexts, the shoulder is located at different, evenly spaced, intervals around the $360°$ circle. In the bottom half of Table 1, we train on three different data sets denoted by the corresponding subgroups of $SO(2)$ that generate the training contexts: C_2, C_4 and C_8. For each of the datasets, we always context 1 in Figure 1 as the base context, and apply various rotations to generate the other training contexts. The C_2 set consists of context 1 together with the subgroup of $180°$ rotations (the $0°$ and $180°$ rotations, resulting in context 1 and context 3 in Figure 1). The set C_4 consists of the $90°$ rotations and the corresponding four training contexts are depicted in Figure 1. Lastly, the C_8 set

consists of the $45°$ rotations, half of which are the $90°$ rotations from Figure 1, and the other half are the $45°$ rotations in between those. Note that for the results of varying ensemble size in the top half of Table 1, we used the C_4 dataset.

The teacher policy is a handcrafted policy ($a = (-2, 2)$ for 12 steps and $a = (2, 2)$ afterwards) that is optimal for the starting pose considered in this setting. The neural network consists of three fully connected hidden layers of size $[64, 64, 32]$ with ReLU activation functions. The policy is distilled with the MSE loss in (6) (which is the same loss as (1) but for vector-valued actions instead of scalar). The exact hyperparameters can be found in Table 4.

Table 4: Hyper-parameters used for the 'Reacher with rotational symmetry' CMDP experiments

'Reacher with rotational symmetry'	
Hyper-parameter	**Value**
Epochs	500
Batch size	6
Learning rate	1×10^{-4}

C.1.3 Figure 2 & Table 2

In the setting from Figure 2 and Table 2, the contexts not only differ in the shoulder location (as described in the subsection above), but also in the starting pose of the robot arm. For testing, a random shoulder location and starting pose are sampled for each episode.

There are four training contexts in this setting, whose shoulder location correspond to the C_4 dataset described above, but whose initial arm poses are sampled randomly by sampling two angles between 0 and 360 degrees for the shoulder and elbow joint. Each seed has its own set of random training poses (but the same shoulder location). The dataset created from these four training contexts is referred to as the *Training Contexts* dataset in Table 2. The *Training Contexts* + C_4 dataset essentially consists of 16 training contexts, four of which are the ones from Training Contexts, and the other 12 are the random poses from the Training Contexts contexts, duplicated for each of the other shoulder locations. Figure 3 illustrates this for an example set of four Training Contexts. The *Training Contexts* + *Random* dataset is the same as the Training Contexts + C_4 set, except that instead of duplicating the random poses from the four Training Contexts, 12 new random poses are sampled.

For this more complicated version of the 'Reacher with rotational symmetry' CMDP, it is much more convoluted to handcraft an optimal policy. Instead, we train an soft actor-critic agent (SAC Haarnoja et al., 2018) with the Stable-Baselines3 (Raffin et al., 2021) implementation on the full context distribution, to get close to an optimal policy for any context. The network for the SAC teacher consists of two fully connected hidden layers of size $[400, 300]$ with ReLU activation functions. The other hyperparameters for the SAC agent can be found in 5. Note that for the results in Table 2, we evaluate single distilled policies ($N = 1$) and we use the same network and hyperparameters for distillation as the experiments for Table 1.

Table 5: Hyper-parameters used for the SAC teacher agent in the 'Reacher with rotational symmetry' CMDP

SAC Teacher	
Hyper-parameter	**Value**
Total timesteps	500 000
Buffer size	300 000
Batch size	256
Discount factor γ	0.99
Gradient steps	64
Train frequency (steps)	64
Target update interval (steps)	1
Target soft update coefficient τ	0.02
Warmup phase	10 000
Share feature extractor	False
Target entropy	auto
Entropy coeff	auto
Use State Dependent Exploration (gSDE)	True
Adam	
Learning rate	5×10^{-4}

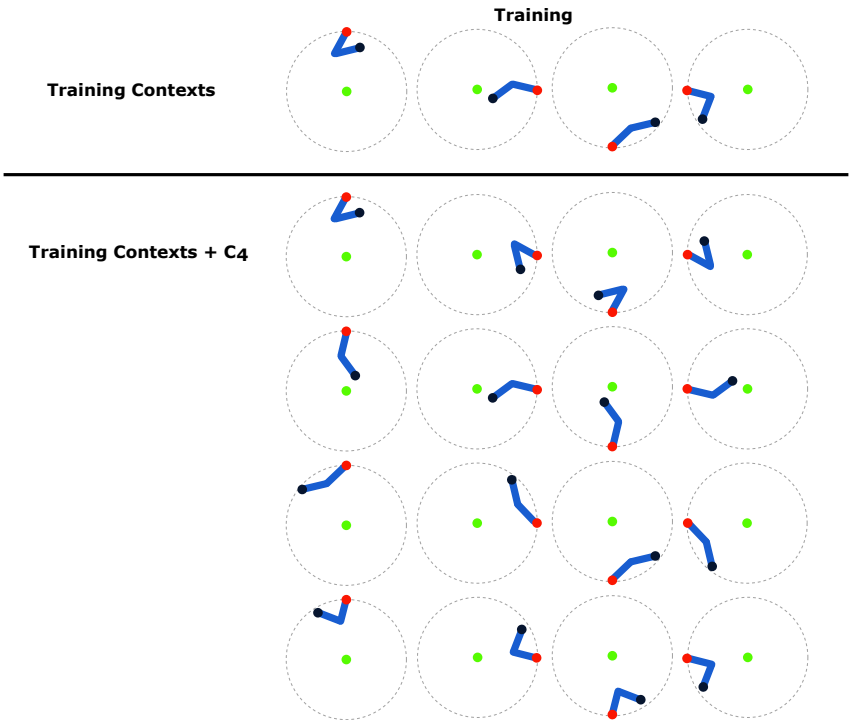


Figure 3: The Training Contexts and Training Contexts + C_4 context sets in the 'Reacher with rotational symmetry' reacher CMDP with varying shoulder location (red) *and* robot arm pose (blue), see Figure 1 for details.

C.2 Four Rooms

In the Four Rooms environment (Figure 4), an agent (red triangle) starts in a random location and facing a random direction, and has to move to the goal location (green square) whilst navigating the doorways connecting the four rooms. We modify the original Minigrid implementation a little bit, by reducing the action space from the default seven (turn left, turn right, move forward, pick up an object,

drop an object, toggle/activate an object, end episode) to only the first three (turn left, turn right, move forward). Moreover, we use a reward function that gives a reward of 1 when the goal is reached and zero elsewhere, which differs from slightly from the default one that gives $1 - 0.9 * (\frac{\text{step count}}{\text{max steps}})$ for reaching the goal. Lastly, our implementation of the Four Rooms environment allows for more control over the context dimensions, allowing for the construction of distinct training and testing sets. Our version of the Four Room environment can be found at `<redacted for review>`.

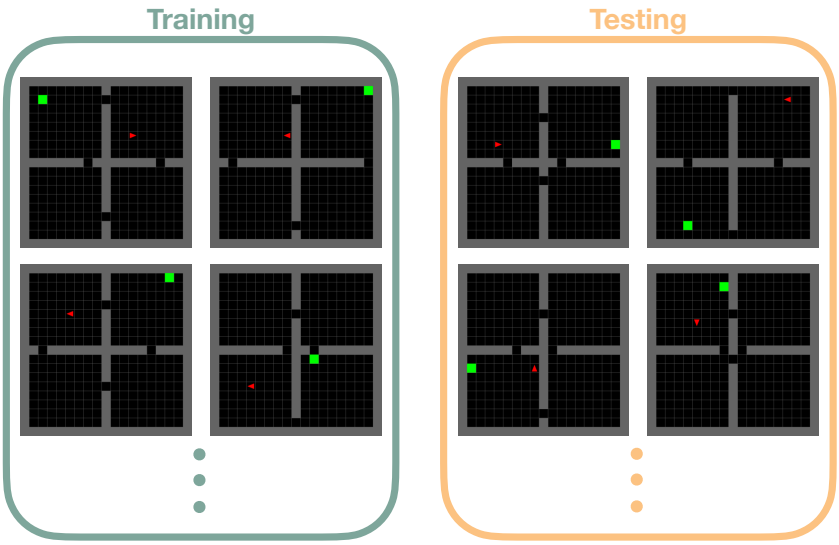


Figure 4: Example of Four Rooms training and testing contexts.

In this environment, the contexts differ in the topology of the doorways connecting the four rooms, the initial location of the agent, the initial direction the agent is facing, and the goal location. The teacher is the PPO+Explore-Go agent from Weltevrede et al. (2025), that is trained on 200 training contexts. We used separate validation and testing sets consisting of unseen contexts of size 40 and 200 respectively. The validation set was used for algorithm development and hyperparameter tuning, and the test set was only used as a final evaluation (and is reported in Table 3). Since the optimal policy in Four Rooms is deterministic, we also evaluate performance of our policies deterministically by always taking the action with maximum probability in a given state.

The *Teacher* dataset used for distillation and behaviour cloning, simply consists of the states encountered by rolling out the stochastic teacher policy in the 200 training contexts until the desired dataset size was reached. We use a dataset size of 500.000, since this was the replay buffer size of the DQN agent from Weltevrede et al. (2025). The *Explore-Go* dataset mimics what a rollout buffer would look like for the PPO+Explore-Go teacher. It is created by running a pure exploration agent (trained as part of the PPO+Explore-Go teacher) for k steps at the beginning of each episode, and afterwards rolling out the stochastic teacher policy until termination of the episode. The number of pure exploration steps k is sampled uniformly from a range $[0, K)$, where $K = 50$ (the same as was used in Weltevrede et al. (2025)). The pure exploration experience is not added to the dataset, only the states encountered by the teacher. The Explore-Go dataset also has size 500.000, but requires additional interactions with the environment (on average $\frac{K}{2} = 25$ steps per episode) that are not added to the dataset. Since the average episode length of the teacher is of similar size, the Explore-Go dataset requires roughly twice the dataset size of additional environment steps to create. Lastly, the *mixed* dataset is a 50/50 mixture of a Teacher dataset of size 250.000, and a dataset created by rolling out the pure exploration policy for 250.000 steps. We generate 20 PPO+Explore-Go teachers, and generate one dataset of each type per teacher (for a total of 20 datasets).

An important thing to note, is that although the states for the distillation and behaviour cloning experiments are the same, the learning targets are not. This has the biggest effect on the Mixed dataset, where the learning targets for behaviour cloning are the actions that were taken to create the dataset, and for distillation, the targets are the PPO+Explore-Go teacher's probabilities in the given state (independent of what action was taken in that state during the creation of the dataset).

The Four Room experiments were executed on a computer with an NVIDIA RTX 3070 GPU, Intel Core i7 12700 CPU and 32 GB of memory. Training of the teacher (PPO agent) would take approximately 2 hours, and a single distillation run would take approximately 10 minutes. The code for our experiments can be found at `<redacted for review>`.

C.2.1 Implementation details

For the distillation experiments (top of Table 3), we used the same architecture as the teacher in Weltevrede et al. (2025). We distil the stochastic teacher policy by regressing on the probabilities (as in Equation 7). We found this to work significantly better than alternative loss functions, since the normalised range for the targets helps the averaging in the ensemble. We tune the distillation hyperparameters by performing a grid search over the following values

- **Learning rate**: $\{1 \times 10^{-4}, 1 \times 10^{-3}, 1 \times 10^{-2}\}$
- **Batch Size**: $\{64, 256, 512, 1024, 2048\}$
- **Epochs**: $\{10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$

We performed the tuning for a single teacher seed by, for each dataset type, splitting the dataset into a training set (sampled from the first 150 training contexts) and validation set (sampled from the other 50 training contexts), and choosing the combination of hyperparameters that minimised the distillation loss on the validation set. The final results are distilled on the full datasets and evaluated in the testing contexts. The final hyperparameters can be found in Table 6.

Table 6: Hyperparameters used for policy distillation in the Four Rooms environment.

Four Rooms Distillation	
Hyper-parameter	**Value**
Teacher	
Epochs	100
Batch size	64
Learning rate	1×10^{-4}
Explore-Go	
Epochs	50
Batch size	512
Learning rate	1×10^{-3}
Mixed	
Epochs	50
Batch size	256
Learning rate	1×10^{-3}

For the behaviour cloning experiments (bottom of Table 3), we used the same architecture as for the distillation experiments. We trained using the logarithmic BC loss for stochastic policies (Equation (8)). We also tuned the behaviour cloning in the same way as the distillation policies above, but with a smaller range for the epochs: $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$, since we found it would overfit much sooner than the distillation experiments. The final hyperparameters can be found in Table 7.

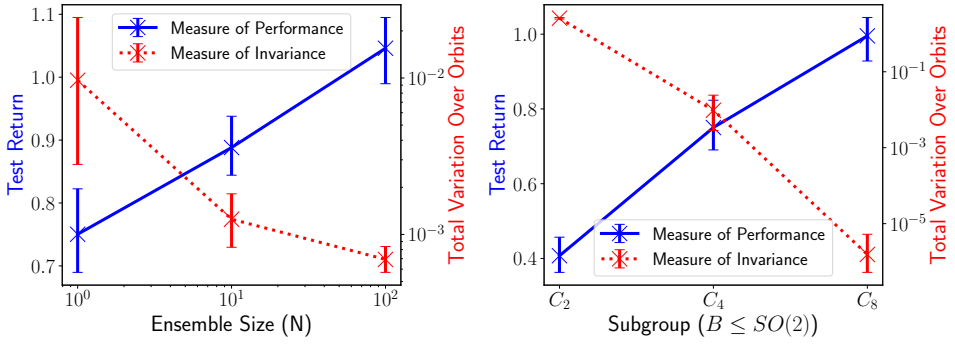
Table 7: Hyperparameters used for behaviour cloning in the Four Rooms environment.

Four Rooms Behaviour Cloning	
Hyper-parameter	**Value**
Teacher	
Epochs	1
Batch size	64
Learning rate	1×10^{-3}
Explore-Go	
Epochs	1
Batch size	64
Learning rate	1×10^{-3}
Mixed	
Epochs	2
Batch size	256
Learning rate	1×10^{-3}

D Additional experiments

D.1 Measure of invariance

In order to identify invariance as a key factor for the increased performance in the 'Reacher with rotational symmetry' environment from Table 1, we will measure how invariant the policy becomes when increasing ensemble or subgroup size. As a measure of invariance we use the variance of the network's output across the group orbit (Kvinge et al., 2022) (for a completely invariant network, this should be zero). In practical terms, we evaluate each policy on all (360) integer degree rotations of the starting state (the orbit) and compute the total variation (trace of the covariance matrix) over the produced outputs. In Figure 5, we see that for both increased ensemble and subgroup size, as the test performance increases, so does the measure of invariance decrease.



(a) (b)

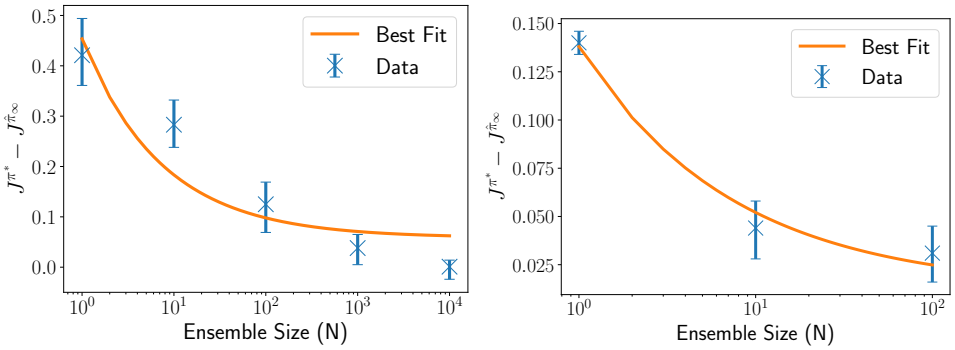
Figure 5: Test return (left axis) compared with the total variation (trace of the covariance matrix) over orbits of the $SO(2)$ group of rotations (right axis) for (a) different ensemble sizes and (b) subgroups $B \leq SO(2)$. The total variation is a measure of how invariant the agent has become with respect to rotations, zero total variation would correspond to perfect invariance. Shown are the mean and 95% confidence intervals over 20 seeds.

D.2 $\frac{1}{\sqrt{N}}$ generalisation bound

In this section, we investigate how well our results fit the $a + \frac{b}{\sqrt{N}}$ form of the generalisation bound from Theory 1. We evaluate different ensemble sizes on the 'Reacher with rotational symmetry' environment from Figure 1, trained on subgroup $B = C_4$ in Table 8. This table includes the results from Table 1, as well as additional results for ensemble sizes $N = 1000$ and $N = 10.000$. In Table 9, we repeat the results for different ensemble sizes distilled on the Explore-Go dataset in the Four Rooms environment from Table 3, with the addition of ensemble size $N = 100$.

In Figure 6, we compare the difference to optimal performance $J^{\pi^*} - J^{\hat{\pi}_\infty}$ with ensemble size N, and plot the best fit to the $a + \frac{b}{\sqrt{N}}$ relation as predicted by our theory. The exact optimal performance J^{π^*} is not necessarily known, but we estimate it by taking the average train performance instead (which seems to have converged on both environments). The best $a + \frac{b}{\sqrt{N}}$ fit is obtained by computing the optimal linear fit between $y = J^{\pi^*} - J^{\hat{\pi}_\infty}$ and $x' = \frac{1}{\sqrt{N}}$ (and then transforming the solution back to $x = N$).

The results in Figure 6 seem to follow the $a + \frac{b}{\sqrt{N}}$ upper bound to some extend. The upper bound is likely not very tight, but the results seem to indicate the bound is also not vacuous.



(a) Reacher with rotational symmetry

(b) Four Rooms

Figure 6: Difference to optimal performance as a function of ensemble size N in the testing contexts for the (a) Reacher with rotational symmetry and (b) Four Rooms environments. Shown are the mean and 95% confidence intervals over 20 seeds from (a) Table 8 and (b) Table 9, together with the best possible fit for the relation $a + \frac{b}{\sqrt{N}}$ as predicted by our theory.

Table 8: Performance of distilled policies in the Illustrative CMDP from Figure 1 for different ensemble sizes N (trained under subgroup $B = C_4$). Shown are the mean and standard deviation for 20 seeds, and in bold are the best returns including those with overlapping 95% confidence intervals.

Ensemble Size N:	N=1	N=10	N=100	N=1000	N=10.000
Train Performance	**1.17 ± 0.004**	**1.17 ± 0.004**	**1.17 ± 0.003**	**1.17 ± 0.003**	**1.17 ± 0.002**
Test Performance	0.75 ± 0.147	0.89 ± 0.107	1.05 ± 0.117	1.13 ± 0.072	**1.17 ± 0.038**

Table 9: Performance of an ensemble (of size N) of policy distillation or behaviour cloning policies on the Explore-Go dataset in the Four Rooms environment. Shown are mean and standard deviation over 20 seeds, and in bold are the best returns including those with overlapping 95% confidence intervals.

Ensemble Size N:	N=1	N=10	N=100
Train Performance	**0.92 ± 0.020**	**0.92 ± 0.019**	**0.92 ± 0.021**
Test Performance	0.78 ± 0.041	**0.88 ± 0.036**	**0.89 ± 0.033**

E Repeated proofs

E.1 Theorem 3 from Maran et al. (2023)

Here we repeat the proof for Theorem 3 in Maran et al. (2023). We mostly change the notation to be consistent with our paper. Note that in Maran et al. (2023) they consider MDPs rather than CMDPs, but since any CMDP (including the testing CMDP $\mathcal{M}|_{C_{test}}$ in a ZSPT setting) is just a special instance of an MDP, the theorem readily applies.

We first introduce a number of definitions and assumptions used in our derivations

Definition 3. *We will call a function f L-Lipschitz continuous if for two metric sets (X, d_x), (Y, d_y), where d_x, d_y are distance metrics, we have*

$$\forall x_1, x_2 \in X, \quad d_y(f(x_1), f(x_2)) \leq L d_x(x_1, x_2).$$

Definition 4. *We define $\|f\|_L$ to be the Lipschitz semi-norm of f, with*

$$\|f\|_L = \sup_{x_1, x_2 \in X, x_1 \neq x_2} \frac{d_y(f(x_1), f(x_2))}{d_x(x_1, x_2)}.$$

Definition 5. *We introduce the Wasserstein distance between probability distributions p and q as*

$$\mathcal{W}(p, q) = \sup_{||f||_L \leq 1} \left| \int f dp - \int f dq \right|.$$

Assumption 1. We assume an (L_T, L_R)-Lipschitz continuous MDP with a metric state and action space and associated distances $d_s (\equiv d_S)$ and $d_a (\equiv d_A)$, for which we have

$$\begin{aligned} \mathcal{W}(T(\cdot|s,a), T(\cdot|\hat{s},\hat{a})) &\leq L_T(d_S(s,\hat{s}) + d_A(a,\hat{a})), \qquad \forall (s,a), (\hat{s},\hat{a}) \in S \times A \\ |R(s,a) - R(\hat{s},\hat{a})| &\leq L_R(d_S(s,\hat{s}) + d_A(a,\hat{a})), \qquad \forall (s,a), (\hat{s},\hat{a}) \in S \times A. \end{aligned}$$

Assumption 2. We assume L_π-Lipschitz continuous policies, which satisfy

$$\mathcal{W}(\pi(\cdot|s), \pi(\cdot|s')) \leq L_\pi d_S(s, s') \quad \forall s, \hat{s} \in S.$$

Note that this definition subsumes deterministic policies.

With this, we can state the theorem in question (Maran et al., 2023)

Theorem 3. *Let π^* be the optimal policy and $\hat{\pi}_\infty$ be the student policy. If the CMDP is (L_T, L_R)-Lipschitz continuous and the optimal and student policies are L_π-Lipschitz continuous, and we have that $\gamma L_T(1 + L_{\hat{\pi}_\infty}) < 1$, then it holds that:*

$$J^{\pi^*} - J^{\hat{\pi}_\infty} \leq \frac{L_R}{(1-\gamma)(1-\gamma L_T(1+L_{\hat{\pi}_\infty}))} \mathbb{E}_{s \sim d^{\pi^*}}[\mathcal{W}(\pi^*(\cdot|s), \hat{\pi}_\infty(\cdot|s))]$$

where $d^{\pi^*}(s) = (1-\gamma)\sum_{t=0}^{\infty} \gamma^t \mathbb{P}(s_t = s|\pi^*, p_0)$ is the γ-discounted visitation distribution and γ the the discount factor.

Proof. From assumptions 1 and 2, it follows that if $\gamma L_p(1+L_\pi) < 1$ the value function Q^π associated with π is L_{Q^π}-Lipschitz continuous (Rachelson and Lagoudakis, 2010) with

$$L_{Q^\pi} \leq \frac{L_R}{1 - \gamma L_T(1 + L_\pi)}$$

The first part of our proof derives performance differences under Lipschitz value functions. We begin with the performance difference Theorem (Kakade and Langford, 2002) stating

$$J^{\pi_1} - J^{\pi_2} = \frac{1}{1-\gamma} \mathbb{E}_{s \sim d^{\pi_1}} \left[\mathbb{E}_{a \sim \pi_1(\cdot|s)}[Q^{\pi_2}(s,a) - V^{\pi_2}(s)] \right]$$

where $V^{\pi_2}(s)$ is the state value function. Focusing on the inner expectation we have

$$\begin{aligned}\mathbb{E}_{a\sim\pi_1(\cdot|s)}[Q^{\pi_2}(s,a) - V^{\pi_2}(s)] &= \int_A (Q^{\pi_2}(s,a) - V^{\pi_2}(s))\pi_1(da|s) \\ &= \int_A Q^{\pi_2}(s,a)\pi_1(da|s) - V^{\pi_2}(s) \\ &= \int_A Q^{\pi_2}(s,a)(\pi_1(da|s) - \pi_2(da|s)).\end{aligned}$$

Now, let $L_s = ||Q^{\pi_2}(s,\cdot)||_L$ be the Lipschitz semi-norm of $Q^{\pi_2}(s,a)$ w.r.t. a and define $g_s(a) = Q^{\pi_2}(s,a)/L_s$ with the property $\|g_s\|_L = 1$. This yields

$$\begin{aligned}\mathbb{E}_{a\sim\pi_1(\cdot|s)}[Q^{\pi_2}(s,a) - V^{\pi_2}(s)] &= \int_A Q^{\pi_2}(s,a)(\pi_1(da|s) - \pi_2(da|s)) \\ &= \int_A g_s(a)L_s(\pi_1(da|s) - \pi_2(da|s)) \\ &= L_s \int_A g_s(a)(\pi_1(da|s) - \pi_2(da|s))\end{aligned}$$

By definition of the Wasserstein distance

$$\mathcal{W}(\pi_1(\cdot|s), \pi_2(\cdot|s)) = \sup_{||g||_L\leq 1} \left| \int_A g(a)(\pi_1(da|s) - \pi_2(da|s)) \right|,$$

such that we have

$$\begin{aligned}\left|\mathbb{E}_{a\sim\pi_1(\cdot|s)}[Q^{\pi_2}(s,a) - V^{\pi_2}(s)]\right| &= \left| L_s \int_A g_s(a)(\pi_1(da|s) - \pi_2(da|s)) \right| \\ &\leq L_s \sup_{\|g\|_L\leq 1} \left| \int_A g(a)(\pi_1(da|s) - \pi_2(da|s)) \right| \\ &= L_s \mathcal{W}(\pi_1(\cdot|s), \pi_2(\cdot|s)).\end{aligned}$$

Now, we recall $L_s = \|Q^{\pi_2}(s,\cdot)\|_L$ and by our assumptions Q^{π_2} is $L_{Q^{\pi_2}}$-Lipschitz continuous such that

$$\begin{aligned}L_s &\leq \sup_{s\in S} ||Q^{\pi_2}(s,\cdot)||_L \\ &\leq \|Q^{\pi_2}\|_L \\ &\leq L_{Q^{\pi_2}}.\end{aligned}$$

Putting these results together, we can obtain

$$J^{\pi_1} - J^{\pi_2} \leq \frac{L_{Q^{\pi_2}}}{1-\gamma} \mathbb{E}_{s\sim d^{\pi_1}}[\mathcal{W}(\pi_1(\cdot|s), \pi_2(\cdot|s))]$$

After setting $\pi_1 = \pi^*$ and $\pi_2 = \hat{\pi}_\infty$ and using that $L_{Q^{\hat{\pi}_\infty}} \leq \frac{L_R}{1-\gamma L_T(1+L_{\hat{\pi}_\infty})}$, we have

$$J^{\pi^*} - J^{\hat{\pi}_\infty} \leq \frac{L_R}{(1-\gamma)(1-\gamma L_T(1+L_{\hat{\pi}_\infty}))} \mathbb{E}_{s\sim d^{\pi^*}}[\mathcal{W}(\pi^*(\cdot|s), \hat{\pi}_\infty(\cdot|s))]$$

□

E.2 Lemma 6.2 from Gerken and Kessel (2024)

Here we repeat the proof for Lemma 6.2 in Gerken and Kessel (2024) in the notation used in this paper.

Lemma 6.2. *Let $\pi_\theta : S \to \mathbb{R}$ be an infinitely wide neural network with parameters θ and with Lipschitz continuous derivatives with respect to the parameters. Furthermore, let $\bar{\pi}_t$ be an infinite ensemble $\bar{\pi}_t(s) = \mathbb{E}_{\theta\sim\mu}[\pi_{\mathcal{L}_t\theta}(s)]$, where the initial weights θ are sampled from a distribution μ and the operator $\mathcal{L}_t$ maps θ to its corresponding value after t steps of gradient descent with respect to a*

MSE loss function. Define the error κ as a measure of discrepancy between representations from the group G and its finite subgroup B:

$$\kappa = \max_{g \in G} \min_{b \in B} ||\psi_S(g) - \psi_S(b)||_{op} \tag{10}$$

The prediction of an infinite ensemble trained with full data augmentation on $B \leq G$ deviates from invariance by

$$\left|\bar{\pi}_t(s) - \bar{\pi}_t\big(\psi_S(g)s\big)\right| \leq \kappa\, C_\Theta(s), \qquad \forall g \in G \tag{11}$$

for any time t. Here $s \in S$ can by any state and C_Θ is independent of g.

Proof. Lets denote a set of states with $\mathcal{D} = \{s_i\}_{i=1}^n$ and a training dataset $\mathcal{T} = \{(s_i, y_i) | \forall s_i \in \mathcal{D}, y_i \in \mathcal{Y}\}$ where $y_i \in \mathcal{Y}$ indicates the target for sample s_i (for example, $y_i = \pi_\beta(s_i)$ for distillation with respect to a teacher π_β). Using the definition of the measure of discrepancy κ and the property of full data augmentation with a finite group from (9), we can write for any training sample $(s_j, y_j) \in \mathcal{T}_B = (\mathcal{D}_B, \mathcal{Y}_B) = \{(\psi_S(b)s, y) | \forall (s, y) \in \mathcal{T}, b \in B\}$ and any $g \in G$ and $b \in B$:

$$||\psi_S(g)s_j - s_{\mathtt{p}_b(j)}|| = ||\psi_S(g)s_j - \psi_S(b)s_j|| \leq ||\psi_S(g) - \psi_S(b)||_{op}||s_j|| < \kappa||s_j|| \,.$$

Additionally, we can use the definition of the mean of the NNGP m_t to write for any $s \in S$

$$\begin{aligned} |\bar{\pi}_t(s) - \bar{\pi}_t\big(\psi_S(g)s\big)| &= |m_t(s) - m_t\big(\psi_S(g)s\big)| \\ &= |(\Theta(s, \mathcal{D}_B) - \Theta(\psi_S(g)s, \mathcal{D}_B))\Theta^{-1}(\mathbb{I} - \exp(-\eta\Theta t))\mathcal{Y}_B| \,. \end{aligned}$$

We can use Lemma 5.2 from Gerken and Kessel (2024), made specific for scalar- and vector-valued functions:

Lemma 5.2. *For scalar- and vector-valued functions, data augmentation implies that the permutation Π_g commutes with any matrix-valued analytical function F involving the NNGP kernel $\mathcal{K}$, the NTK Θ and their inverses:*

$$\Pi(g)F(\Theta, \Theta^{-1}, \mathcal{K}, \mathcal{K}^{-1}) = F(\Theta, \Theta^{-1}, \mathcal{K}, \mathcal{K}^{-1})\Pi(g)$$

where $\Pi(g)$ denotes the permutation matrix applying the permutation $\mathtt{p}_g$ associated with g to each training point.

to show that:

$$\begin{aligned} \Theta(s, \mathcal{D}_B)\Theta^{-1}(\mathbb{I} - \exp(-\eta\Theta t))\mathcal{Y}_B &= \Theta(s, s_i)\Theta^{-1}_{ij}(\mathbb{I} - \exp(-\eta\Theta t))_{jk}y_k \\ &= \Theta(s, s_i)\Theta^{-1}_{ij}(\mathbb{I} - \exp(-\eta\Theta t))_{jk}y_{\mathtt{p}_b(k)} \\ &= \Theta(s, s_{\mathtt{p}_b^{-1}(i)})\Theta^{-1}_{ij}(\mathbb{I} - \exp(-\eta\Theta t))_{jk}y_k \end{aligned}$$

where we also used invariance of the labels: $\Pi(b)\mathcal{Y}_B = \mathcal{Y}_B$. Plugging this into the expression above:

$$\begin{aligned} |\bar{\pi}_t(s) - \bar{\pi}_t\big(\psi_S(g)s\big)| &= |(\Theta(s, s_{\mathtt{p}_b^{-1}(i)}) - \Theta(\psi_S(g)s, s_i))\Theta^{-1}_{ij}(\mathbb{I} - \exp(-\eta\Theta t))_{jk}y_k| \\ &= |(\Theta(s, s_{\mathtt{p}_b^{-1}(i)}) - \Theta(s, \psi_S^{-1}(g)s_i))\Theta^{-1}_{ij}(\mathbb{I} - \exp(-\eta\Theta t))_{jk}y_k| \end{aligned}$$

Now, we bound the following:

$$\begin{aligned} \Delta\Theta(s', s, \bar{s}) &= |\Theta(s', s) - \Theta(s', \bar{s})| \\ &= \left| \sum_{l=1}^{L} \mathbb{E}_{\theta\sim\mu}\left[\left(\frac{\partial \pi_\theta(s')}{\partial \theta^{(l)}}\right)^\top \left(\frac{\partial \pi_\theta(s)}{\partial \theta^{(l)}} - \frac{\partial \pi_\theta(\bar{s})}{\partial \theta^{(l)}}\right)\right]\right| \\ &\leq ||s - \bar{s}|| \sum_{l=1}^{L} \mathbb{E}_{\theta\sim\mu}\left[\left|\left(\frac{\partial \pi_\theta(s')}{\partial \theta^{(l)}}\right)^\top \cdot L(\theta^{(l)})\right|\right] \\ &= ||s - \bar{s}||\hat{C}(s') \end{aligned}$$

where $L(\theta^{(l)})$ is the Lipschitz constant of $\partial_{\theta^{(l)}}\pi_\theta$. Finally, using the triangle inequality:

$$\begin{aligned} |\bar{\pi}_t(s) - \bar{\pi}_t\big(\psi_S(g)s\big)| &\leq \hat{C}(s)\sqrt{\sum_i ||s_{\mathtt{p}_b^{-1}(i)} - \psi_S(g)s||^2}\sqrt{\sum_i (\sum_{j,k} \Theta^{-1}_{ij}(\mathbb{I} - \exp(-\eta\Theta t))_{jk}y_k)^2} \\ &\leq \kappa\hat{C}(s)\sqrt{\sum_i ||s_i||^2}\sqrt{\sum_i (\sum_{j,k} \Theta^{-1}_{ij}(\mathbb{I} - \exp(-\eta\Theta t))_{jk}y_k)^2} = \kappa C_\Theta(s) \end{aligned}$$

□

E.3 Lemma B.4 from Gerken and Kessel (2024)

Here we repeat the proof for Lemma B.4 in Gerken and Kessel (2024) in the notation used in this paper.

Lemma B.4. *The probability that the deep ensemble $\bar{\pi}_t$ and its Monte-Carlo estimate $\hat{\pi}_t$ differ by more than a given threshold δ is bounded by*

$$\mathbb{P}\big[|\bar{\pi}_t(s) - \hat{\pi}_t(s)| > \delta\big] \leq \sqrt{\frac{2}{\pi}} \frac{\sigma_s}{\delta} \exp\bigg(- \frac{\delta^2}{2\sigma_s^2}\bigg),$$

where we have defined

$$\sigma_s^2 := Var(\hat{\pi}_t)(s) = \frac{\Sigma_t(s)}{N}$$

where $\Sigma_t(s)$ is the NNGP variance and N is the finite ensemble size.

Proof. In the infinite width limit, the ensemble members for our Monte-Carlo estimator $\hat{\pi}_t$ are i.i.d. random variables drawn from a Gaussian distribution with mean $\bar{\pi}_t$ and variance Σ_t. Therefore, the probability of deviation for a given threshold δ is given by

$$\mathbb{P}\big[|\bar{\pi}_t(s) - \hat{\pi}_t(s)| > \delta\big] = \frac{2}{\sqrt{2\pi}\sigma_s} \int_\delta^\infty \exp\bigg(- \frac{x^2}{2\sigma_s^2}\bigg) \mathrm{d}x \,,$$

where $\sigma_s^2 = \frac{\Sigma_t(s)}{N}$ is the variance of the Monte-Carlo estimator $\hat{\pi}_t$. With a change of integration variable $u = \frac{t}{\sigma_s\sqrt{2}}$, and using the fact that $1 \leq \frac{2u}{2\min(u)}$ for $u \geq \min(u)$, we get:

$$\mathbb{P}\big[|\bar{\pi}_t(s) - \hat{\pi}_t(s)| > \delta\big] = \frac{2}{\sqrt{\pi}} \int_{\frac{\delta}{\sqrt{2}\sigma_s}}^\infty \exp\bigg(- u^2\bigg) \mathrm{d}u \leq \frac{1}{\sqrt{\pi}} \frac{\sqrt{2}\sigma_s}{\delta} \int_{\frac{\delta}{\sqrt{2}\sigma_s}}^\infty (2u) \exp\bigg(- u^2\bigg) \mathrm{d}u \,.$$

Finally, this integral evaluates to

$$\mathbb{P}\big[|\bar{\pi}_t(s) - \hat{\pi}_t(s)| > \delta\big] \leq \sqrt{\frac{2}{\pi}} \frac{\sigma_s}{\delta} \exp\bigg(- \frac{\delta^2}{2\sigma_s^2}\bigg)$$

□

NeurIPS Paper Checklist

1. **Claims**
   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?
   Answer: [Yes]
   Justification: The claims are supported by the theoretical and experimental contributions in sections 4 and 5.
   Guidelines:
   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.
2. **Limitations**
   Question: Does the paper discuss the limitations of the work performed by the authors?
   Answer: [Yes]
   Justification: The limitations of the theoretical assumptions are investigated and discussed in Section 6.
   Guidelines:
   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.
3. **Theory assumptions and proofs**
   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?
   Answer: [Yes]
   Justification: The assumptions are clearly listed in Section 4 and the proof can be found in Appendix B.
   Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The experimental details can be found in Appendix C including a link to the code.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code can be found at `https://github.com/MWeltevrede/distillation-after-training`.

   Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The experimental details can be found in Appendix C

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: For all results, standard deviation is reported and their significance is evaluated by checking for overlapping 95% confidence intervals.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).

- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The amount of compute required is mentioned in Appendix C.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We conform to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: The paper concerns fundamental research for which it is difficult to reason about any societal impacts.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We do not have data or models that have a high risk for misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Any original creators or owners are properly credited and licences are respected.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Code for the experiments is released with the camera ready version.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: We do not have any experiments with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: This paper does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.