
Distilling Policy Distillation

Wojciech M. Czarnecki
DeepMind

Razvan Pascanu
DeepMind

Simon Osindero
DeepMind

Siddhant M. Jayakumar
DeepMind

Grzegorz Świrszcz
DeepMind

Max Jaderberg
DeepMind

Abstract

The transfer of knowledge from one policy to another is an important tool in Deep Reinforcement Learning. This process, referred to as *distillation*, has been used to great success, for example, by enhancing the optimisation of agents, leading to stronger performance faster, on harder domains [26, 32, 5, 8]. Despite the widespread use and conceptual simplicity of distillation, many different formulations are used in practice, and the subtle variations between them can often drastically change the performance and the resulting objective that is being optimised. In this work, we rigorously explore the entire landscape of policy distillation, comparing the motivations and strengths of each variant through theoretical and empirical analysis. Our results point to three distillation techniques, that are preferred depending on specifics of the task. Specifically a newly proposed *expected entropy regularised distillation* allows for quicker learning in a wide range of situations, while still guaranteeing convergence.

1 Introduction

Reinforcement Learning (RL) and in particular Deep Reinforcement Learning (DRL) has shown great success in recent years, from agents playing Atari games from raw-pixel inputs [16, 7], to beating professional players in GO [30] or learning dexterous robotic manipulation [19]. However, obtaining these levels of

performance can often require an almost prohibitively large amount of experience to be acquired by the agent in order to learn [25]. Consequently, there is great interest in techniques that can allow for knowledge transfer; that is enabling the training of agents based on already trained policies [24] or human examples [1]. One of the most successful techniques for knowledge transfer is that of *distillation* [10, 24, 23], where an agent is trained to match the state-dependent probability distribution over actions provided by a teacher. Examples include allowing us to: train otherwise untrainable agent architectures [5]; speed up learning [26]; build stronger policies [23]; drive multi-task learning [2, 24]. Analogous techniques are widely used in supervised learning to achieve model compression [10, 11, 22], reparametrisation for inference speedup [18], and joint co-training of multiple networks [34].

Although the high-level formulation of distillation in RL is simple, one can find dozens of different mathematical formulations used in practice. For example: sometimes trajectories are sampled from a teacher [24], sometimes from the student [13] or a mixture [23]; some authors use a KL divergence between teacher and student distribution [9] while others look at KL between entire trajectory probabilities [32]. A primary goal of this paper is to provide a roadmap of these different ideas and approaches, and then to perform a step-by-step comparison of them, both mathematically and empirically. This allows us to construct a set of guidelines for deciding which specific distillation approach might best fit a particular problem.

The main contributions of this paper can be summarised as follows: In Section 3.1 we provide a proof that commonly used distillation with trajectories sampled from the student policy does not form a gradient vector field, and while it has convergence guarantees in simple tabular cases, it can oscillate as soon as one introduces rewards to the system. We show simple methods of recovering the gradient vector field prop-

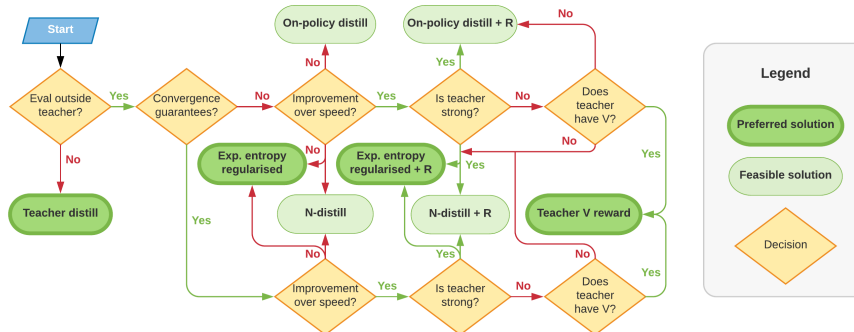


Figure 1: A distillation method selection diagram, based on the results from this paper. Note, that in practise there might be many other factors affecting such decisions, but this serves as a good initial rule of thumb. Method references: **Teacher distill**: [24, 23, 9]; **On-policy distill**: [23, 20, 13]; **On-policy distill+R**: [26, 5]; **Entropy regularised**: [32, 27]; **N-distill**: Section 3.1; **Exp. entropy regularised**: Section 3.3; **Teacher V reward**: Section 4.2. See Table 1 for methods equations and properties.

erty. In Section 3.2 we perform empirical evaluation of different control policies, showing when and why it is beneficial to use student-driven distillation. In Section 4 we analyse the actor-critic setup, in which one has access to a teacher’s value function, V , in addition to its policy, and discuss how V may also be used for distillation. We empirically evaluate all the above techniques in thousands of random MDPs. Finally, based on all the results combined from our mathematical analyses and empirical evaluation, we propose effective new distillation variants and provide a rule of thumb decision tree, Fig. 1.

2 Preliminaries

Throughout the paper we assume that we are working with Markov Decision Processes, and will now outline our notation. We have a finite set of states, \mathcal{S} , a finite set of possible actions, \mathcal{A} , an agent policy $\pi : \mathcal{S} \rightarrow \Delta^{|\mathcal{A}|}$ which outputs distribution over actions in each state. Agents interact with an environment at time t by sampling actions $a_t \sim \pi(\tau_t)$, and the environment transitions to a new state according to unknown transition dynamics $\tau_{t+1} \sim T(\tau_t, a_t)$ and produces rewards $r_t \sim r(\tau_t, a_t)$. Each $\tau_t \in \mathcal{S}$ is the state encountered at time t in the trajectory τ . We use \mathbb{E}_π to denote an expectation over the distribution of trajectories, τ , generated by agent when interacting with the environment using policy π . Under this notation the typical goal of reinforcement learning is to find $\pi^* = \arg \max_\pi \left[\mathbb{E}_\pi \left[\sum_{t=1}^{|\tau|} \gamma^{t-1} r_t \right] \right]$, where $\gamma \in [0, 1]$ is discount factor. For simplicity we use $\gamma = 1$ in most of our theoretical results, though the proofs can trivially be extended to arbitrary $\gamma \leq 1$. For the empirical results we generally use $\gamma = .99$, see the Ap-

pendix A. We consider the general problem of extracting knowledge from a *teacher policy*, π , and transferring it to a different *student policy*, π_θ , using trajectories, $\tau = \{\tau_1, a_1, r_1, \dots, \tau_{|\tau|}, a_{|\tau|}, r_{|\tau|}\}$, sampled from interactions between a control policy, q_θ , and the unknown environment.

All proofs are provided in the Appendices C and D.

3 Policy distillations

Through the rest of this paper we consider update rules for θ (parameters of the student policy π_θ) which are proportional to:

$$\mathbb{E}_{q_\theta} \left[\sum_{t=1}^{|\tau|} -\nabla_\theta \log \pi_\theta(a_t | \tau_t) \widehat{R}_t + \nabla_\theta \ell(\pi_\theta, V_{\pi_\theta}, \tau_t) \right], \quad (1)$$

for $\widehat{R}_t = \sum_{i=t}^{|\tau|} \widehat{r}_i = \sum_{i=t}^{|\tau|} \widehat{r}(\pi_\theta, V_{\pi_\theta}, \tau_i, a_i, \tau_{i+1}, a_{i+1}, r_i)$ and a choice of q, ℓ and \widehat{r} that define a specific instance of a distillation technique (see Table 1 for a list of examples). In this equation, ℓ can be seen as a form of auxiliary loss [12] responsible for policy alignment at the current step, while \widehat{r} can be viewed as a reward term that combines extrinsic and intrinsic components [21] and thus is responsible for long-term alignment. Note, we assume undiscounted objectives and episodic RL, but analogous analysis can be performed for the discounted case.

Focusing on update rules rather than simply losses may seem to add unnecessary complexity, however one of the crucial outcomes of our work is to show that the update rules involved in certain distillation methods do not have corresponding loss functions. Consequently we must make an explicit distinction between

Table 1: A comparison of various ways of defining distillation between the teacher (π) and student (π_θ) policies. q_θ is the sampling (control) policy. ∇ column denotes whether the update rule prescribed is a valid gradient vector field. $H^\times(p(s)||q(s))$ denotes Shannon’s cross entropy between two distributions over actions $-\mathbb{E}_{a\sim p(s)}[\log q(a|s)]$. * For proof see Theorem 1. ** For proof see Theorem 2. Methods below the mid line are introduced in this paper – usually as modifications of known techniques to address specific issues identified. Each of the techniques but *Teacher distill* and *Teacher V reward* has a corresponding $+R$ version where \hat{r}_i is replaced with $\hat{r}_i + r_i$.

name	q_θ	$\ell(\pi_\theta, V_{\pi_\theta}, \tau_t)$	\hat{r}_i	is ∇ ?	Loss
Teacher distill	π	$H^\times(\pi(\tau_t) \pi_\theta(\tau_t))$	0	yes [1]	$\mathbb{E}_\pi[\sum_t H^\times(\pi(\tau_t) \pi_\theta(\tau_t))]$
On-policy distill	π_θ	$H^\times(\pi(\tau_t) \pi_\theta(\tau_t))$	0	no*	does not exist*
Entropy regularised	π_θ	0	$\log \pi(a_i \tau_i)$	yes [4]	$\mathbb{E}_{\pi_\theta}[\sum_t -\log \pi(a_t \tau_t)]$
N-distill	π_θ	$H^\times(\pi(\tau_t) \pi_\theta(\tau_t))$	$-H^\times(\pi(\tau_{i+1}) \pi_\theta(\tau_{i+1}))$	yes**	$\mathbb{E}_{\pi_\theta}[\sum_t H^\times(\pi(\tau_t) \pi_\theta(\tau_t))]$
Exp. entropy regularised	π_θ	$H^\times(\pi_\theta(\tau_t) \pi(\tau_t))$	$\log \pi(a_{i+1} \tau_{i+1})$	yes**	$\mathbb{E}_{\pi_\theta}[\sum_t -\log \pi(a_t \tau_t)]$
Teacher V reward	π_θ	0	$r_i + V_\pi(\tau_{i+1}) - V_{\pi_\theta}(\tau_i)$	yes**	$\mathbb{E}_{\pi_\theta}[\sum_t r_t]$

update rules, and losses which may be used to derive update rules.

3.1 Control policy

Many RL distillation frameworks set up knowledge transfer as a supervised learning problem [23, 24, 9, 33], by following updates in the direction of: $\mathbb{E}_\pi \left[\sum_{t=1}^{|\tau|} \nabla_\theta H^\times(\pi(\tau_t)||\pi_\theta(\tau_t)) \right]$, with Monte Carlo estimates for the expectation based on trajectories derived from the teacher policy, π . However, since then, several publications [20, 5, 26] have reported better empirical results when trajectories are sampled from the student instead, i.e. by following updates in the direction of: $\mathbb{E}_{\pi_\theta} \left[\sum_{t=1}^{|\tau|} \nabla_\theta H^\times(\pi(\tau_t)||\pi_\theta(\tau_t)) \right]$. Note that in this form of update the gradient operator is under an expectation wrt. the same set of variables that it operates upon. Consequently it is not clear if this process will even converge, and thus the benefits of using such updates are also unclear.

In this section we analyse and prove the following properties: (i) For tabular policies, provided q_θ guarantees a non-zero probability of sampling each state visited by the teacher, the dynamics will converge. In particular $q_\theta = \pi_\theta$ with a softmax policy satisfies this property; (ii) In general, updates like this do not form gradient vector fields; (iii) If one adds reward optimisation to the system, the dynamics can cycle and never converge; (iv) A reward-based correction term can be added to ensure convergence (and with such a correction, the updates do correspond to proper gradient vector field); (v) There is a trade-off between the speed of convergence and the fidelity of the behaviour replication, which can be controlled by q_θ .

We begin by proving a general theorem about on-policy non-gradients. In principle, it is very similar to the notion of *compatibility* of a value function and

the policy [31], or can be seen as a generalisation of *incompatibility* towards other possible trajectory level losses $\ell(\tau|\theta)$ (e.g. $\ell(\tau|\theta) = \sum_{t=1}^{|\tau|} \ell(\pi(\tau_t)||\pi_\theta(\tau_t))$).

Theorem 1. *Let us assume that $g(\theta) = \mathbb{E}_{\pi_\theta}[\nabla_\theta \ell(\tau|\theta)]$ is differentiable and there does not exist $\alpha_\tau \in \mathbb{R}$ such that $\nabla_\theta \ell(\tau|\theta) = \alpha_\tau \nabla \pi_\theta(\tau)$ almost everywhere. Then $g(\theta)$ is not a gradient vector field of any function.*

The assumption about the existence of α_τ is equivalent to the compatibility criterion [31], and thus it shows that incompatible value functions do not create valid gradient vector fields. This is complementary to the result that compatible value functions provide convergence to the optimal policy. If we choose $\ell(\tau|\theta) = \sum_{t=1}^{|\tau|} H^\times(\pi(\tau_t)||\pi_\theta(\tau_t))$ we recover the on-policy distillation updates used in techniques such as kickstarting [26] and Mix&Match [5]. In this setting there is no corresponding α_τ that simply rescales policy logits, and thus as a consequence of Theorem 1 we see that naive distillation with student-generated trajectories does not form a gradient vector field. We also note that exactly the same proof shows that the entropy penalty [15] $\ell(\tau|\theta) = -\sum_{t=1}^{|\tau|} H(\pi(\tau_t))$, commonly used in actor critic algorithms, also results in updates that do not correspond to a valid gradient vector field.

Having seen that these commonly used updates do not correspond to a valid gradient vector field, a natural question to ask is whether this is necessarily problematic. For example – the updates used in Q-learning are not gradient steps either, but Q-learning still provides a convergent iterative scheme. We address the question of what can be said about the dynamical system emerging from this sort of distillation, and for a simple tabular setup we can show that indeed this is not an issue:

Proposition 1. *Using an update rule of the form $\mathbb{E}_{\pi_\theta}[\sum_{t=1}^{|\tau|} \nabla_\theta \ell(\pi(\tau_t), \pi_\theta(\tau_t))]$ for a strongly stochas-*

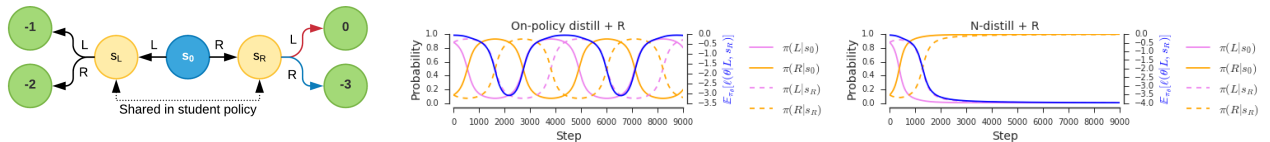


Figure 2: **Left:** An MDP providing an example of non-convergence of on-policy distillation, when distilling from a teacher which prefers to go right (blue arrow) when in s_R over going left (red arrow) by putting a loss $\ell(\theta|s_R) = -4\pi_\theta(R|s_R)$. For simplicity it does not provide any learning signal in the remaining states. The initial state is coloured blue, L and R are actions, and green states are terminal states with rewards inside. **Center and Right:** Evolution of the policy is shown, with clear oscillation for on-policy distill and convergence for n-distill. The student policy is parameterised with sigmoids, shares parameter for s_L and s_R , and trained with true expected returns. Detailed proof of oscillations is provided in the Appendix C.

*tic*¹ student policy, with episodic finite state-space MDPs and tabular policies, provides convergence to the teacher policy over all reachable states for the loss function ℓ , provided the optimiser used can minimise $\ell(a, b)$ wrt. b , for any a in the domain of ℓ , and $\ell(a, b)$ reaches minimum at $\ell(a, a)$.

However, despite this positive result, we can also show that even in the episodic MDP case one can break convergence if we introduce rewards. The counterexample, visualised in Fig. 2 and described in detail in the Appendix C, relies on a teacher that can discriminate between some states that the student cannot. It leads to an oscillation – the student policy will never converge even after infinitely many steps.

There are multiple possible ways to construct on-student-policy distillation learning methods similar to the ones used in practice, but which do provide update rules that are gradient vector fields². One such way is to start from the objective suggested by the update rule component, namely: $\mathbb{E}_{\pi_\theta} \left[\sum_{t=1}^{|\tau|} \ell(\pi(\tau_t) \| \pi_\theta(\tau_t)) \right]$. Then we compute its gradient using the log-derivative trick, analogously to how the KL-regularised RL objective is derived [27] or how Stochastic Computation Graphs are obtained [28]; doing so gives the update direction:

$$\mathbb{E}_{\pi_\theta} [\nabla_\theta \ell(\tau|\theta)] + \mathbb{E}_{\pi_\theta} [\nabla_\theta \log \pi_\theta(\tau) \ell(\tau|\theta)].$$

As we can see, the gradient vector field is composed of two expectation terms. The first term corresponds

¹Meaning that each for each action a , parameters θ and state s , $\pi_\theta(s)[a] > 0$.

²It is worth noting that the typical trick of importance sampling is not viable here. First, it is unclear what sampling distribution to correct with respect to – one can choose any distribution that is independent of θ , thus it could be teacher policy, but also say – a uniform one. Second, mathematically this simply leads to degeneration to optimisation of the corresponding loss, such as teacher distill rather than on-policy method.

to the 1-step on-policy distillation setup discussed so far. The second term corresponds to the standard RL objective if $-\ell$ plays the role of the reward function. This simple derivation allows us to prove the following:

Theorem 2. *In order to recover the gradient vector field property for 1-step on-policy distillation updates with any loss $\ell(\pi(\tau_t) \| \pi_\theta(\tau_t))$, one can add an extra reward term $\hat{r}_t = -\ell(\pi(\tau_{t+1}) \| \pi_\theta(\tau_{t+1}))$. Analogously if the loss is of the form $\mathbb{E}_{a \sim \pi_\theta} \hat{\ell}(\pi(\tau_t))$ then the correction is of form $-\hat{\ell}(\pi(\tau_{t+1}))$.*

Since this is a gradient vector field, it can be safely composed with reward based updates without losing any convergence properties. As one can see on the right of Fig. 2 applying this correction to On-policy distill+R (and thus creating the *N-distill+R*), leads to convergence and minimisation of the loss, as expected.

Given the potential convergence issues with the naive updates from Equation 1, particularly when also considering reward from the environment as highlighted by our counterexample, it begs the question: Why does following the student’s policy when performing distillation typically lead to better empirical results? Our main hypothesis is that, *if convergent*³, it provides more robust policies wrt. trajectories sampled from the student.

This follows the general machine learning principle of training in the same regime as that which we expect to encounter during test time. In particular, after distillation, the goal is usually to either evaluate a student agent when it is generating its own actions, or to allow the student agent to continue training on its own. Therefore what matters is an expectation wrt. π_θ , and not wrt. π . Consequently performing distillation “on-policy” wrt. student trajectories leads to less of a distribution-shift between training and testing phases.

³In practise researchers often force convergence by learning rate annealing, early stopping etc. so the issues highlighted here may often be masked.

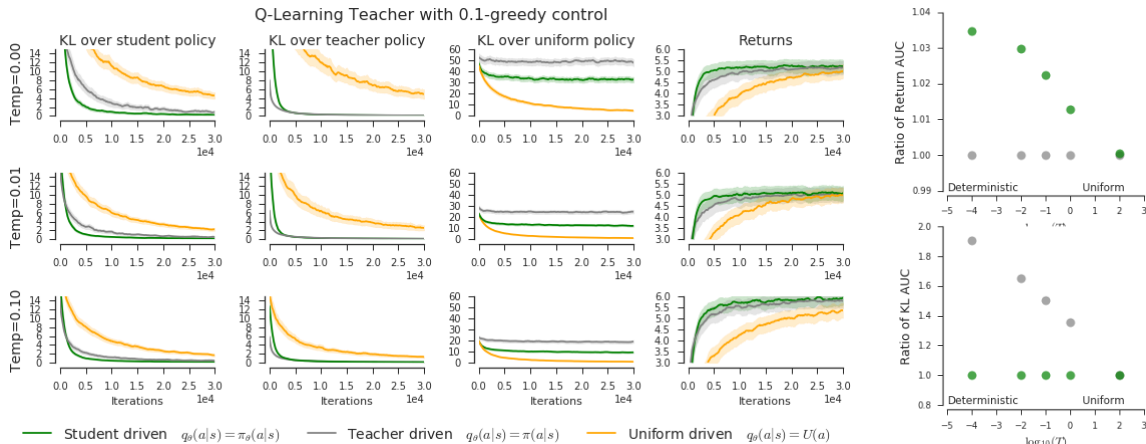


Figure 3: **Left:** Learning curves averaged over 1k grid worlds when distilling Q-Learning teacher with various control policies (colors) and temperatures (rows). We report KL over various sampling distributions z_θ (columns) and returns when following teacher policy. Shaded region represents 0.95 confidence intervals, estimated as $1.96 \cdot$ standard error of the mean. **Right:** Relation between student-driven distillation speed-up (measured as ratio of areas under reward/KL curves) and teacher determinism.

Another motivating argument is that if the teacher is almost deterministic, then it visits a relatively small fraction of the state space, even though during training it might have built a policy to deal with other situations too. When using π during distillation, the student will not have the opportunity to replicate the teacher’s behaviour in these states, since it will not visit them. In general, after distillation, $\pi \neq \pi_\theta$ and they diverge quickly in complex environments or over long trajectories. Again, on-student-policy distillation avoids these issues, especially initially – many states will be visited that are not normally encountered under the teacher policy. Consequently one can expect better replication of the teacher, when measured in the entire state space.

The main observations of this section are: (i) on-policy 1-step distillation updates do not form gradient fields, and when mixed with environment rewards can lead to non-convergent behaviour; (ii) distillation using student-generated trajectories replicates the teacher policy in more states that are relevant under the student’s behaviour distribution. The following empirical section highlights the effects that different choices for the control-policy have in practice.

3.2 Empirical evaluation of the control policy

We consider teacher driven [24], student driven [20] and fixed (uniform) control policies. We define a distribution over grid world tasks, where we randomly place walls, terminating and rewarding states in 20×20 2D grid worlds (see Appendix A.2 for a detailed description of the MDP generating procedure), and agents

are capable of moving in 4 directions. There is a fixed probability of terminating each episode, such that we end up with bounded (undiscounted) returns. We sample 1k MDPs like this, and distill for 30k optimisation steps using various control policies q_θ . Since we have proven that in the tabular case we can use distillation based on per-step cross-entropy, H^\times , this is the loss we are minimising, using a gradient based update to the underlying logits.

We use teachers trained with Q-Learning and ϵ -greedy policies (ϵ set to 0.1, full details provided in the Appendix A.4) and observe how different types of control policy affect the distillation loss. We measure $\mathbb{E}_{z_\theta} \left[\sum_{t=1}^{|\tau|} H^\times(\pi(\tau_t) \parallel \pi_\theta(\tau_t)) \right]$ for various choices of z_θ , which can be different from q_θ used for distillation.

As predicted by our theoretical analysis, student-driven distillation brings benefits wrt. the loss computed over student trajectories. In other words, if one cares about how closely the student behaviour matches the teacher behaviour when the student agent is allowed to generate experience on its own, student-driven distillation optimises this quantity well, with the gap disappearing as the teacher is more and more uniform (entropic), see Fig. 3. Similarly, matching of the teacher policy outside of typically visited states (measured by z_θ being uniform) is also much better when following a student-driven policy. The best result, in terms of whole state space, is achieved when using uniform distribution for q_θ , but this is a control setting which does not scale to larger state or action spaces. And when we compare this control setting to the student-based setting we see that it converges ex-

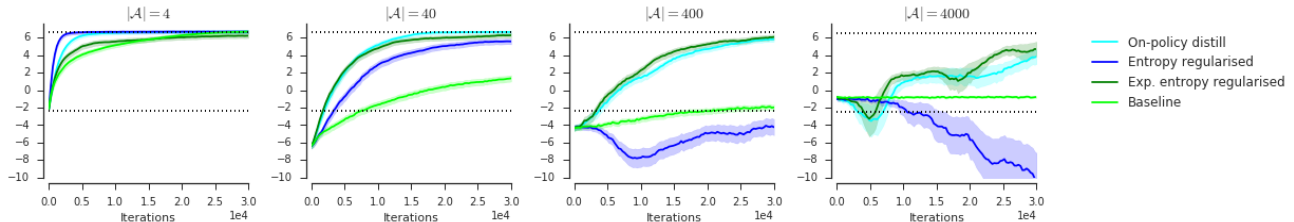


Figure 4: Learning curves averaged over 1k random MDPs with $|\mathcal{A}|$ actions, out of which 4 are movement actions and the remaining ones do not affect the movement of the agent, but simply make exploration hard. Plots show the failure mode of the intrinsic reward only based distillation, and how their expected version fixes it.

tremely slowly even in these scenarios.

Similarly, in terms of the returns obtained by the agent, student driven distillation sees the fastest learning progress, and needs on average $3\times$ less steps than teacher driven distillation (and around $10\times$ less than uniform driven) to recover full teacher performance. This is of crucial importance, since distillation in RL is typically a first step in a larger training procedure in which the rewards obtained by the student policy are to be maximised with potentially further training. In order to be useful for these applications, one typically seeks to obtain highly rewarding policies as rapidly as possible. The only criterion under which teacher driven distillation works more effectively is, somewhat obviously, the expected KL under trajectories generated from teacher distribution. However this is an artificial scenario which is rarely true in practice.

To summarise, student-driven distillation provides significant improvements in terms of empirical results over teacher-driven distillation. While one could heuristically drive the switch between the two [23], the pure student-driven method seems to be strong enough to use solely, as long as a proper loss is being used, which we discuss in the next section. Therefore, in the remainder of this paper we focus on student driven distillation $q_\theta = \pi_\theta$.

3.3 Empirical evaluation of various updates

An important choice is the selection of which method we use to update the student policy given the trajectory and actions suggested by the teacher policy. There are two popular approaches here: one is to try to maximise the probability of the trajectory generated by the student under the teacher policy [32, 27], and the other is to frame the learning as a per-timestep supervised learning problem, defining the loss at each timestep to be the cross entropy between the teacher’s and student’s distributions over actions [2, 26, 5]

There are two aspects worth discussing. Firstly, in the entropy regularised setup, where we use $H^\times(\pi_\theta||\pi)$

the student would be considered a prior, while the teacher a posterior, however in the distillation setup with $H^\times(\pi||\pi_\theta)$ the teacher is the prior and the student the posterior. Secondly, the cross entropy regularised approach (which tries to minimise the cross entropy between whole trajectories distributions $\mathbb{E}_{\pi_\theta}[\sum_t -\log \pi(a_t|\tau_t)]$) can be absorbed in the reward channel, without any ℓ being used. Using only the reward signal can suffer from very high variance in the gradient estimator, for example when the action space is large. As one can see from Fig. 4, as we increase the size of the action space from four actions up to 4k actions (by adding actions that do not move the agent) the speed of entropy regularised distillation drastically collapses, while traditional distillation still works.

One can reduce the variance by splitting the entropy term into a 1-step update expressed by ℓ , and incorporate the remaining updates through \hat{r} . This technique, denoted as *expected entropy regularised*, indeed recovers performance of traditional distillation. While this is not a new objective as such, but rather a different estimator, it is, to our best knowledge a novel method, which strictly dominates popular alternatives.

The direction chosen for the cross entropy has a very simple intuitive explanation. If one uses $H^\times(\pi||\pi_\theta)$ then one tries to replicate π , however if one uses $H^\times(\pi_\theta||\pi)$ then one tries to find a deterministic policy, which puts all probability mass on the most probable action of π (see the Appendix C for a proof). If the cost is changed to be the KL divergence instead, this issue is eliminated for finite action spaces, however for continuous control it is a matter of mean vs median seeking techniques [14] (see the Appendix C for more detail). Depending on the MDP, both methods can be beneficial, and of course for almost deterministic teachers – they actually are equivalent.

To summarise, these experiments demonstrate that across the space of possible student-driven distillation approaches the most reliable method, both mathematically and empirically, is our proposed *expected entropy regularised distillation*. It has three key benefits: (i)

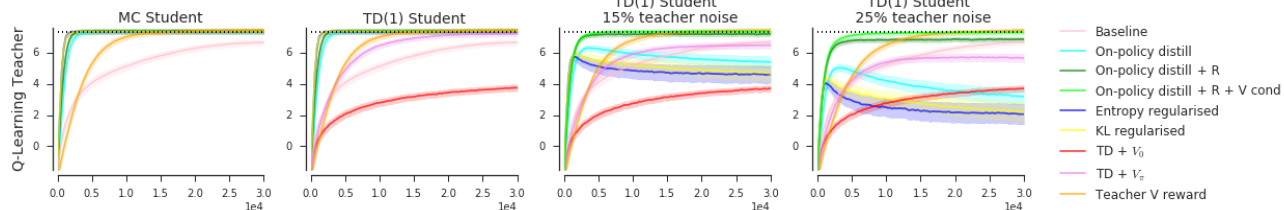


Figure 5: Learning curves obtained from averaging 1k training runs on randomly generated MDP grid worlds. The gradual decrease in reward when distilling from a sub-optimal Q-Learning teacher with distillation methods that enforce full *policy cloning* comes from the fact that the teacher is purely deterministic – while being closer to it initially helps, once the student replicates all the wrong state decisions perfectly its reward start to decrease.

it creates a valid gradient vector field; (ii) it does not suffer from high-variance of the estimate typically used in similar methods [32]; and (iii) it directly maximises the probability of the student produced trajectories under the teacher policy, as opposed to the *n-distill* method, which looks at maximising the probability of being in states where the student and teacher agree. Consequently, it combines the best elements of various similar techniques, while avoiding their drawbacks.

4 Policy distillation from Actor-Critic

In practice, while pure policy based methods remain useful [23] in RL, the Actor-Critic framework [15, 29, 7] has risen to prominence in recent years. Consequently we now shift our attention towards distillation strategies that make use of value critics, denoted by $V_\pi(s) = \mathbb{E}_\pi[\sum_t r_t | \tau_1 = s]$. The availability of this additional knowledge, in the form of $V_\pi(s)$, allows us to better leverage imperfect teachers, as we can begin to estimate how much to trust them.

For example, let’s assume that we have ground-truth access to V_{π_θ} and V_π , and consider a loss of the form: $\ell(\theta, s) = H^\times(\pi(s) || \pi_\theta(s)) [V_\pi(s) - V_{\pi_\theta}(s)]_{>0}$, where $[x]_{>0} = 1$ iff $x > 0$. This can be seen as an action-independent version of the Generalised Policy Improvement technique [4]. As a result we can easily prove that after converging, our agent will be at least as strong as its teacher, independent of the initial returns of π_θ .

Proposition 2. *For S_1 being a distribution over initial states, if we have $\forall s \in S \ell(\theta^*, s) \leq H(\pi(s))$ then $\mathbb{E}_{s \sim S_1}[V_{\pi_{\theta^*}}(s)] \geq \mathbb{E}_{s \sim S_1}[V_\pi(s)]$.*

With techniques like this we can use teachers which do make mistakes. One such experiment, where we randomly flip the most probable action of the teacher policy in a given percentage of states it visits, is illustrated in Fig. 5. As before we generate 1k random MDPs, and empirically estimate V_π by sampling 100 trajectories after a given fraction of state-action pairs

have been modified. As one can see, methods that fully replicate the teacher (in this case a Q-Learning based one, with 25% noise) end up saturating in sub-optimal solutions – around 2-4 points (Fig. 5 blue and cyan curves). If we add true rewards to the learning system, we improve, but still saturate around 6.5 points (Fig. 5 dark green curve). With the value function based gating described in Proposition 2 we recover the full performance of the teacher with approximately 7 points (Fig. 5 lime curve). As expected, the usefulness of this approach depends on the quality of the teacher and the accuracy of the value function estimators.

4.1 Using the teacher’s critic to bootstrap

Yet another way of using the teacher’s knowledge is to use its value functions, V_π , instead of the student’s, V_{π_θ} , for bootstrapping – so that the usual actor-critic TD(1) update direction: $\nabla \log \pi_\theta(a_t | \tau_t) [r(a_t, \tau_t) + V_{\pi_\theta}(\tau_{t+1})]$ instead becomes $\nabla \log \pi_\theta(a_t | \tau_t) [r(a_t, \tau_t) + V_\pi(\tau_{t+1})]$, which we will refer to as TD + V_π . Note, that it is no longer optimising for the true return since the teacher’s value function is not with respect to the student’s policy. Consequently, it is not obvious if the resulting update is a gradient vector field, nor whether the result of the updates has meaningful fixed points. We can show that for simple cases it is a valid update rule:

Proposition 3. *Assume that we are given the true value V_π of the teacher policy π , for a finite state size MDP, then optimising using $\mathbb{E}_{\pi_\theta}[\sum_t \nabla \log \pi_\theta(a_t | \tau_t) [r(a_t, \tau_t) + V_\pi(\tau_{t+1})]]$ converges to a policy with $\mathbb{E}_{s \sim S_1} V_{\pi_\theta}(s) \geq \mathbb{E}_{s \sim S_1} V_\pi(s)$ for S_1 being the distribution of initial states.*

As the previous experiments show (Fig. 5), bootstrapping from the teacher leads to a better performance than just replicating it (which empirically confirms the theoretical claim of an improvement), but at the same time it is not solving the original problem, and so the method will still saturate early on, if the teacher is not strong enough (e.g.: see the poor results of TD + V_0

where the teacher just predicts 0 value everywhere).

4.2 Using the critic as an intrinsic reward

Another possible way to use the teacher’s critic is to define an intrinsic- (or shaping-) reward based on its value $\hat{r}_t^V := V_\pi(\tau_{t+1}) - V_\pi(\tau_t) + r_t$, which will provide reward proportional to the increase in value over the last action. Note, that now we have to add true reward, as value can decrease when the rewarding state is encountered and we do not want to penalise our agent for actually obtaining the reward.

It is easy to conclude, that such an update rule is going to guarantee convergence to the policy which has the maximal value⁴ while at the same time it is not forcing complete policy cloning even if π is the optimal policy (which is the property of any potential based intrinsic reward [17, 3, 6]).

Although the presence of such a value-function based shaping reward does not affect the optimal solutions achieved, the learning dynamics are affected. If the teacher is strong (close to optimal) then it is going to help convergence speed; if it is weak (obtains low returns), it can slow down training. We can formalise it with the following proposition

Proposition 4. *Let us assume the teacher is an optimal policy for the given MDP, then for each action a_t that would lead to a deviation from the optimal path, it will get an immediate penalty, meaning that $r_t^V < r_t$, while following any of the optimal paths leads to $r_t^V = r_t$.*

In a symmetric counterpoint to Proposition 4, if our teacher is the worst possible policy (minimising returns rather than maximising) we get the opposite proposition – where one receives immediate penalties for doing anything but the worst possible actions. While this still does not affect the solution of learning, it will slow down training significantly.

As a concrete example of these concepts, we can use a simple grid world where one can only go left or right, while starting in the middle of a corridor of length $2T + 1$, and rewards +1 and -1 are placed at the right and left ends of the corridor respectively. We define an optimal teacher $\pi_{\text{opt}}(R|s_i) = 1.0$ and an adversarial teacher $\pi_{\text{bad}}(L|s_i) = 1.0$ (but we set $V_{\pi_{\text{bad}}}(a|s) := V_{\pi_{\text{good}}}(a|s)$, thus creating adversarial value function). When distilling from an optimal policy, distillation techniques based on policy cloning recover optimal scores very quickly (Fig. 6). Knowledge transfer using shaping rewards from the teacher’s value

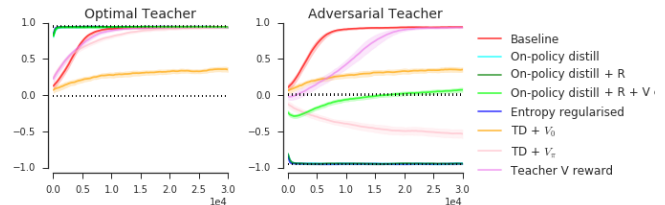


Figure 6: Results of distilling an optimal teacher and an adversarial (minimising reward) teacher from the chain-structured MDP described in Section 4.2 with $T = 5$. See results for other T in Appendix Fig. 9.

function leads to much slower progress, but it still outperforms our baseline training without any distillation. However, once we switch to the adversarial teacher, all the policy cloning approaches fail, while the *Teacher V reward* still learns. It is also worth noting that bootstrapping from the teacher’s value function fails, as it is only guaranteed to improve upon the teacher (which it does), not to solve the original task.

5 Conclusions

In this paper we sought to highlight some of the strengths, weaknesses, and potential mathematical inconsistencies in different variants of distillation used for policy knowledge transfer in reinforcement learning. In particular, we suggest a unifying view of many different techniques which allows them to be compared and understood side-by-side. We provide both theoretical analyses, as well as large scale empirical studies on synthetic MDPs, which allows us to provide a prescriptive suggestion of best-practices for distillation in different settings. The synthesis of these findings is summarised in the flowchart in Figure 1.

A key contribution of our work is to demonstrate that many widely used methods do not correspond to valid gradient vector fields, and thus may be susceptible to non-convergent learning dynamics. However, armed with our insights, we are able to suggest modifications to these approaches which address some of these dynamical issues. In particular we found that expected entropy regularised distillation to be the most reliable formulation, both theoretically and based on our empirical results. Also, if available, the critic of the teacher can be used to deal with imperfect teachers.

While the synthetic MDPs used in this work were crucial for exploring effectively different hypothesis, as future work, an open question is how these approaches behave empirically on large-scale, real-world problems, using function approximators like deep neural networks. Mathematically understanding the role of the function approximator is also left as future work.

⁴ $\mathbb{E}_{\pi_\theta}[\sum_{t=1}^{|\tau|} \hat{r}_t^V] = \mathbb{E}_{\pi_\theta}[\sum_{t=1}^{|\tau|} V_\pi(\tau_{t+1}) - V_\pi(\tau_t) + r_t] = \mathbb{E}_{\pi_\theta}[-V_\pi(\tau_1) + \sum_{t=1}^{|\tau|} r_t] = \mathbb{E}_{\pi_\theta}[\sum_{t=1}^{|\tau|} r_t] + \text{const.}$

Acknowledgements

We would like to thank Jane Wang, Yee Whye Teh and Nicolas Heess for many insightful discussions and feedback on this work as well as the whole DeepMind team for their support.

References

- [1] Abbeel, P. and Ng, A. Y. (2004). Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1. ACM.
- [2] Arora, H., Kumar, R., Krone, J., and Li, C. (2018). Multi-task learning for continuous control. *arXiv preprint arXiv:1802.01034*.
- [3] Asmuth, J., Littman, M. L., and Zinkov, R. (2008). Potential-based shaping in model-based reinforcement learning. In *AAAI*, pages 604–609.
- [4] Barreto, A., Dabney, W., Munos, R., Hunt, J. J., Schaul, T., van Hasselt, H. P., and Silver, D. (2017). Successor features for transfer in reinforcement learning. In *Advances in neural information processing systems*, pages 4055–4065.
- [5] Czarnecki, W. M., Jayakumar, S. M., Jaderberg, M., Hasenclever, L., Teh, Y. W., Osindero, S., Heess, N., and Pascanu, R. (2018). Mix&match-agent curricula for reinforcement learning. *International Conference on Machine Learning*.
- [6] Devlin, S. and Kudenko, D. (2012). Dynamic potential-based reward shaping. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 433–440. International Foundation for Autonomous Agents and Multiagent Systems.
- [7] Espeholt, L., Soyer, H., Munos, R., Simonyan, K., Mnih, V., Ward, T., Doron, Y., Firoiu, V., Harley, T., Dunning, I., Legg, S., and Kavukcuoglu, K. (2018). Scalable distributed deep-rl with importance weighted actor-learner architectures. *International Conference on Machine Learning*.
- [8] Gangwani, T. and Peng, J. (2017). Genetic policy optimization. *arXiv preprint arXiv:1711.01012*.
- [9] Ghosh, D., Singh, A., Rajeswaran, A., Kumar, V., and Levine, S. (2017). Divide-and-conquer reinforcement learning. *arXiv preprint arXiv:1711.09874*.
- [10] Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.
- [11] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.
- [12] Jaderberg, M., Mnih, V., Czarnecki, W. M., Schaul, T., Leibo, J. Z., Silver, D., and Kavukcuoglu, K. (2017). Reinforcement learning with unsupervised auxiliary tasks. *International Conference on Learning Representations*.
- [13] Lin, K., Wang, S., and Zhou, J. (2017). Collaborative deep reinforcement learning. *arXiv preprint arXiv:1702.05796*.
- [14] Minka, T. et al. (2005). Divergence measures and message passing. Technical report, Technical report, Microsoft Research.
- [15] Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937.
- [16] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529.
- [17] Ng, A. Y., Harada, D., and Russell, S. (1999). Policy invariance under reward transformations: Theory and application to reward shaping. In *ICML*, volume 99, pages 278–287.
- [18] Oord, A. v. d., Li, Y., Babuschkin, I., Simonyan, K., Vinyals, O., Kavukcuoglu, K., Driessche, G. v. d., Lockhart, E., Cobo, L. C., Stimberg, F., et al. (2017). Parallel wavenet: Fast high-fidelity speech synthesis. *arXiv preprint arXiv:1711.10433*.
- [19] OpenAI, Andrychowicz, M., Baker, B., Chociej, M., Jozefowicz, R., McGrew, B., Pachocki, J., Petron, A., Plappert, M., Powell, G., Ray, A., Schneider, J., Sidor, S., Tobin, J., Welinder, P., Weng, L., and Zaremba, W. (2018). Learning dexterous in-hand manipulation.
- [20] Parisotto, E., Ba, J. L., and Salakhutdinov, R. (2015). Actor-mimic: Deep multitask and transfer reinforcement learning. *arXiv preprint arXiv:1511.06342*.
- [21] Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. (2017). Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning (ICML)*, volume 2017.
- [22] Polino, A., Pascanu, R., and Alistarh, D. (2018). Model compression via distillation and quantization. In *International Conference on Learning Representations*.

- [23] Ross, S., Gordon, G., and Bagnell, D. (2011). A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635.
- [24] Rusu, A. A., Colmenarejo, S. G., Gulcehre, C., Desjardins, G., Kirkpatrick, J., Pascanu, R., Mnih, V., Kavukcuoglu, K., and Hadsell, R. (2015). Policy distillation. *arXiv preprint arXiv:1511.06295*.
- [25] Salimans, T., Ho, J., Chen, X., Sidor, S., and Sutskever, I. (2017). Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*.
- [26] Schmitt, S., Hudson, J. J., Zidek, A., Osindero, S., Doersch, C., Czarnecki, W. M., Leibo, J. Z., Kuttler, H., Zisserman, A., Simonyan, K., et al. (2018). Kickstarting deep reinforcement learning. *arXiv preprint arXiv:1803.03835*.
- [27] Schulman, J., Chen, X., and Abbeel, P. (2017). Equivalence between policy gradients and soft q-learning. *arXiv preprint arXiv:1704.06440*.
- [28] Schulman, J., Heess, N., Weber, T., and Abbeel, P. (2015a). Gradient estimation using stochastic computation graphs. In *Advances in Neural Information Processing Systems*, pages 3528–3536.
- [29] Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. (2015b). Trust region policy optimization. In *International Conference on Machine Learning*, pages 1889–1897.
- [30] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016). Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484.
- [31] Sutton, R. S., McAllester, D. A., Singh, S. P., and Mansour, Y. (2000). Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063.
- [32] Teh, Y., Bapst, V., Czarnecki, W. M., Quan, J., Kirkpatrick, J., Hadsell, R., Heess, N., and Pascanu, R. (2017). Distral: Robust multitask reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 4496–4506.
- [33] Yin, H. and Pan, S. J. (2017). Knowledge transfer for deep reinforcement learning with hierarchical experience replay. In *AAAI*, pages 1640–1646.
- [34] Zhang, Y., Xiang, T., Hospedales, T. M., and Lu, H. (2017). Deep mutual learning. *arXiv preprint arXiv:1706.00384*, 6.