Ensemble Policy Distillation in Deep Reinforcement Learning

Yuxiang Sun
Computer Science and Engineering Department
University of South Carolina
syuxiang@email.sc.edu

Pooyan Fazli
Department of Computer Science
San Francisco State University
pooyan@sfsu.edu

Abstract

Policy distillation in deep reinforcement learning transfers the knowledge learned by a large teacher model to a compact student model, which reduces the inference time and power consumption. However, the compression ratio and the long training time are not always satisfactory. One promising approach is for the teacher's training and student's distillation to occur simultaneously, so that the latest learned policy is distilled in real time. However, an intrinsic problem arises when the teacher provides unstable supervision, as this may misdirect the distillation process and lead to failure. Until now, only few research works have addressed the problem of instability and distillation performance. In this work, we propose a policy distillation mechanism that applies *ensemble distillation* in a new way, which makes more high-quality and reliable supervisions available for the student to realize full distillation. In addition, the validity of ensemble distillation has been demonstrated for the improvement of generalization, which enhances the student model's robustness. We verify our algorithm in the OpenAI Atari game domain. The results show that the proposed approach achieves nearly full distillation and even greater performance on some tasks.

Introduction

Deep reinforcement learning (DRL), as the combination of reinforcement learning and deep neural networks has demonstrated its potential in a variety of challenging tasks, such as games (Mnih et al. 2015), navigation (Mirowski et al. 2016), and manipulation (Lillicrap et al. 2015). However, a DRL model is generally computationally expensive, which impedes its application in capacity- and power-constrained devices such as drones and cellphones. Policy distillation proves to be an effective way to address this problem, where a light-weight student model is trained under the supervision of a pre-trained teacher model to learn efficient policies (Rusu et al. 2015; Parisotto, Ba, and Salakhutdinov 2015). However, there are still unresolved problems in the existing approaches, such as prolonged training and distillation time or inadequate distillation performance with very small compression ratios. Policy distillation in real time is able to reduce the time significantly by distilling the policy when the teacher is training, but the effectiveness of the approach is diminished by the teacher's unstable performance. Since a student model is vulnerable to fluctuations in the teacher's supervision due to the student model's relatively lower approximation ability, it is hard for the student to return to the right track and learn the optimal policy once directed in a wrong way. Until now, the underlying mechanism of this instability and uncertainty in policy distillation has not been explored widely.

To address the problem of unreliable teacher supervision in policy distillation, we introduce an ensemble learning method in which several teacher models generate more reliable supervision for the student under the mechanism of real-time policy distillation (Sun and Fazli 2019). Ensemble learning has previously shown its strength in other knowledge distillation tasks (Asif, Tang, and Harrer 2019) and improved robustness in adversarial cases (Shen et al. 2019) addressing the problems of overfitting and high variance. In addition, reverse KL divergence is specifically adopted in this work to reduce the uncertainty in the teacher model. Combining these two techniques, our results demonstrate that a highly compressed student model is able to effectively master teachers' skills and even surpass these skills to some degree.

In summary, the contributions of the work are as follows:

- We introduce an ensemble method for policy distillation and verify its effectiveness in the OpenAI Atari game domain.
- We compare forward KL divergence and reverse KL divergence and verify that the latter is a better match for policy distillation in deep reinforcement learning.

Background

This work is based on one benchmark experiment DQN (Deep Q-learning (Mnih et al. 2015)), where a teacher model is constructed with two identical networks: a prediction network and a target network. Existing policy distillation approaches usually involve two steps. In the first step, for the training phase, a teacher model is trained to converge to an optimal policy by minimizing the Bellman error recursively. In this step, the i-th iteration is written as in Equation 1 (Mnih et al. 2015). For convenience of illustrating, we call it a DQN loss.

$$\mathcal{L}_{DQN} = \mathbb{E}_{s,a}[\underbrace{r_{i+1} + \gamma \underset{a}{max} Q(s_{i+1}, a; \theta_i^-)}_{target} - \underbrace{Q(s_i, a; \theta_i)}_{prediction}]^2. \tag{1}$$

In the second step, for the distillation phase, the trained teacher plays the game under the learned policy and collects data for the student. Given the same data sampled from the replay buffer, both teacher and student models are required to output the confidence over all actions under individual policy independently. The student uses only the prediction network to minimize the distribution regression loss, such as KL divergence, formulated in Equation 2:

$$\mathcal{L}_{KL}(\theta^{(S)}) = \boldsymbol{p}_i^{(T)} ln(\frac{\boldsymbol{p}_i^{(T)}}{\boldsymbol{p}_i^{(S)}}), \tag{2}$$

where $\boldsymbol{p}_i^{(T)} = softmax(\frac{\boldsymbol{q}_i^{(T)}}{\tau})$ and $\boldsymbol{p}_i^{(S)} = softmax(\frac{\boldsymbol{q}_i^{(S)}}{\tau})$. The terms $\boldsymbol{q}_i^{(T)}$ and $\boldsymbol{q}_i^{(S)}$ are q-value vectors approximated by the prediction network in the teacher model and the student model, respectively. $\theta^{(S)}$ represents weights in the student model. The softmax function is used to obtain the normalized confidence of the actions chosen. The temperature τ measures how soft or hard targets should be to can convey the most useful information.

Different from other approaches in the literature (Rusu et al. 2015; Parisotto, Ba, and Salakhutdinov 2015), we use the real-time policy distillation mechanism, where distillation occurs during instead of after the teacher's training. In other words, the latest policy being learned by the teacher is transferred immediately to the student. The student is expected to be smart enough to learn as quickly as possible, with no time for hesitance or consolidation of previous knowledge.

Uncertainty Reduction via Reverse KL Divergence

One commonly used distillation loss function is KL divergence, also called forward KL divergence (FKL) as Equation 2. Its reverse form is reverse KL divergence (RKL). In this paper, RKL is recommended instead of FKL due to the match between the property of RKL and the traits of DRL, formulated as Equation 3,

$$\mathcal{L}_{RKL}(\theta^{(S)}) = \boldsymbol{p}_i^{(S)} ln(\frac{\boldsymbol{p}_i^{(S)}}{\boldsymbol{p}_i^{(T)}}). \tag{3}$$

In analyzing the different properties of these two KL divergence losses (Sun and Fazli 2019), one interesting observation (Kingma and Welling 2013; Doersch 2016) guides the decision of which KL divergence is an ideal candidate. FKL holds the *zero-avoiding* property (Malinin and Gales 2019), in that it tends to allocate nonzero probability for each action in the student model if the teacher's output distribution is indistinguishable. This confuses the student's decision-making. Specifically, in Equation 2, it is the teacher who weights the term $ln(\frac{\boldsymbol{p}_i^{(T)}}{\boldsymbol{p}_i^{(S)}})$ by its confidence term $\boldsymbol{p}_i^{(T)}$ to determine the loss value. In contrast, RKL is *zero-forcing*, in which the student weights the total KL loss via its confidence $\boldsymbol{p}_i^{(S)}$ in Equation 3. This allows the student to ignore the supervision and allocate zero or low probability for some actions, as long as the student is very confident in taking one certain action as the best one, and effectively prevents the transfer of *harmful knowledge* from the teacher.

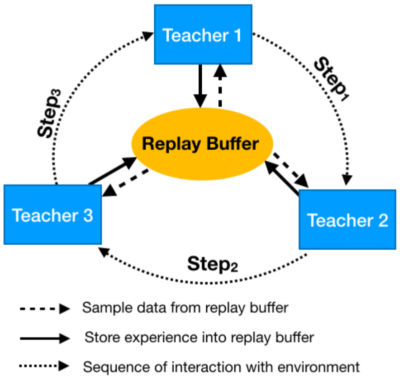
Teacher 1
Step1
Step3
Replay Buffer
Teacher 3
Teacher 2
Step2
Sample data from replay buffer
Store experience into replay buffer
Sequence of interaction with environment

Figure 1: Construction of Replay Buffer via Multiple Teachers

Since RL is learning without ground truth to select the best action, the corresponding decisions are sensitive to noise. From our observation, the confidences over all actions are nearly equal in the early stages of the DQN training or when the policy is degrading, accordingly, teachers' supervision is informative and uncertain. In this case, FKL still pushes the student to imitate the teacher's decision as much as possible, even if the student is holding better policy. Consequently, the student's policy will be overridden by force. However, RKL authorizes autonomy to a student to make a decision on its own and also reduces uncertainty via distillation. Extremely, as the decomposition of RKL gradient in the paper *Real-time* policy distillation (Sun and Fazli 2019), a totally random supervision only contributes a constant in gradient update, proportional to $1/N$, N is the number of optional actions.

Distillation from Ensemble Distribution

A noisy and weak RL model is easily driven to local optima for various reasons, such as inappropriate model initialization, the learning rate schedule, and the amount of positive knowledge contained in the sampled experience. Therefore, the performance may fluctuate drastically during training, and policy degradation may occur at different training points with different random seeds. This could result in catastrophic guidance for the student in the framework of real-time policy distillation, because undesirable supervision may totally reshape the student's parameter landscape and even override the student's previous knowledge, which is not recoverable for some tasks. Generally, the teacher model is sufficiently flexible and more able to abandon a sub-optimal policy and return to the right track than a student because of the larger number of parameters and computing redundancy. Lowering the risk of unreliable supervision as a target for the student's training is nontrivial in real-time policy distillation to make full distillation.

Enlightened by the idea of ensemble learning (Zhang and Wang 2016; Siddique, Alam, and Adeli 2019), we constructed ensemble teacher models in the training phase, with the purpose of shrinking the performance variance and generating consensus among teachers. Before discussing our implementation details, the arrangement of these teachers to collect data should be noted. Specifically, we must determine which teacher's experience should be stored into the replay buffer, an important component that improves data reusability and prevents oscillation in a convergence scenario by sampling with uniform probability in the original DQN. Nair et al. (2015) introduced multiple actors interacting independently on different servers to explore the state space widely from different paths. They constructed a merged replay buffer to force the agent to confront diverse situations by learning from all actors' demonstrations in the early stages, which also enhanced its robustness. For simplicity in our case, all teachers are built up in the same network, play in a single game, and make decisions alternately. The teachers' experiences are stored in the same replay buffer with equal opportunity. Figure 1 depicts how we organize the sequence of multiple teachers' interactions with the environment for each playing turn. Only one of the teachers is required to output the best action and store the corresponding experience into the replay buffer at each step. During the training time, the sampled data is fed into all models, and all teachers optimize policy using gradient descent in the DQN loss. Meanwhile, the student updates the policy via distribution regression by minimizing the following loss,

$$\mathcal{L}_{Ensemble}(\theta^{(S)}) = \sum_{j=1}^{M} \boldsymbol{p}_i^{(S)} ln(\frac{\boldsymbol{p}_i^{(S)}}{\boldsymbol{p}_i^{(T_j)}}), \qquad (4)$$

where M is the number of teacher models. The pseudocode for the procedure is shown in Algorithm 1. The experience data sampled from the replay buffer is denoted as $\mathcal{D}_{replay} = \{s_i, a_i, r_i, s_{i+1}\}$, and a *distillation dataset* is constructed for the student, denoted as $\mathcal{D}_{distill} = \{\mathcal{D}_{replay}, \quad p_i^{(T_{1:M})}\}$. The second part $p_i^{(T_{1:M})}$ represents the set of output distributions derived from the ensemble models.

There are two benefits from the alternate decision-making. First, each teacher gets a chance to learn from the others' demonstrations, which is helpful for correcting the policy bias to some extent, especially when the teacher model is misled to local optima. Second, it is less likely for all teacher models to fall into local optima at the same time. Thus, even if one teacher undergoes poor policy and its collected data contain negative knowledge, this situation is mitigated by other teachers' data, and the risk of too much unsuccessful experience in the replay buffer is reduced.

Some existing ensemble distillation methods (Kuncoro et al. 2016) train an extra component *voting* by weighting each teacher's KL loss to select the best predictions that further improve the supervision quality. In our approach, RKL naturally retains the attribute of voting, even though it uses a simple averaging operation to obtain a synthesized opinion. The reason is that, if one teacher model is playing under an undesirable policy and outputs more random predictions, then the proportion of RKL loss between it and the student model will be bound by the student's weight, which lowers that teacher's contribution to the final opinion. In contrast, if only the student is stuck in local optima, it will accept supervision from teachers by weighting with its equal weighting term; thus, its learning will be corrected immediately. In the extreme case, if both student and teacher models make uniformly random decisions, the RKL loss between them would remain relatively constant, causing no contributions to loss minimization.

Algorithm 1 Ensemble Policy Distillation

```
$M$: number of teachers
$I$: number of total iterations
$\mathcal{C}$: update frequency of target network
Initialize replay buffer $\mathcal{D}_{replay}$.
Initialize the teacher models and the student model with random weights for both the prediction and target networks.
while $1 < i < I$ do
    for $j = 1, M$ do
        Teacher $j$ interacts with environment and stores experience into replay buffer;
        Data is randomly sampled from replay buffer $\mathcal{D}_{replay}$ and feed into all models;
        Teachers update weights $\theta^{(T_{1:M})}$ according to DQN loss function computation in Equation 1;
        Distillation training set $\mathcal{D}_{distill}$ is constructed;
        Student updates weights $\theta^{(S)}$ according to RKL loss function computation as equation 4;
        Weights from prediction network are copied to target network every $\mathcal{C}$ steps.
    end for
    $i \leftarrow i + M$
end while=0
```

Experiment Setup

To assess the performance of the proposed distillation approach, we selected seven games (Rusu et al. 2015) from the Atari domain. There is one teacher model with size Teacher=$\{32, 64, 64, 512\}$, one student model (Rusu et al. 2015) student1=$\{16, 16, 16, 64\}$, and one smaller student model student2=$\{8, 8, 16, 64\}$. The first three numbers represent the number of filters in the convolutional layers, and the last is the number of neurons in the fully-connected layer. The students' corresponding compression ratios with respect to the total number of parameters are 3.7% and 1.7% respectively. We followed the training and evaluation settings in the original DQN (Mnih et al. 2015), introducing 30 random actions in the beginning of each episode, evaluating every single epoch by playing up to 30 episodes. In our training phase, all models were trained for 80 epochs ($250,000$ steps per epoch), and evaluated sequentially. The temperature in softmax was uniformly set to 0.1 as one appropriate hyperparameter for most games to sharpen the targets from the ensemble from our empirical study, it is potentially optimized

Table 1: Distillation results for "Student1". **PD**: Policy Distillation (Rusu et al. 2015); **Single**: with only one teacher's supervision; **Ensemble**: proposed method with 3 teacher models; **FKL**: forward KL divergence; **RKL**: reverse KL divergence; Note: in the case of ensemble, the average of teachers' scores is used to compute the percentage.

	PD	Single-RKL				Ensemble-FKL				Ensemble-FKL			
	%	*Score*		*Percentage*		*Score*		*Percentage*		*Score*		*Percentage*	
	Max	**Max**	**Mean**	**Max**	**Mean**	**Max**	**Mean**	**Max**	**Mean**	**Max**	**Mean**	**Max**	**Mean**
Pong	96.9	14.0	11.8	106.8	96.5	**15.3**	**14.1**	**113.8**	**110.5**	14.9	13.8	110.3	108.4
Breakout	78.6	**31.5**	**16.2**	128.7	105.1	29.2	16.0	**162.4**	**122.7**	21.7	14.8	120.6	112.8
Enduro	117.1	**419**	288	104.8	102.7	354	**317**	130.8	**200.7**	371	316	**132.6**	200.0
Ms.Pacman	96.2	**1553**	303	98.1	32.0	1497	**1041**	**153.4**	**108.5**	1116	936	128.5	97.8
Qbert	107.9	1150	698	89.7	71.5	**4614**	**3295**	**129.7**	**110.2**	4093	3069	115.1	95.7
Seaquest	**137.4**	1805	1498	103.2	103.5	**1934**	**1682**	134.5	**135.5**	1884	1570	132.3	126.5
Riverraid	84.1	3700	2867	106.0	97.6	**4584**	**4142**	**120.9**	**114.8**	4202	3732	110.9	103.5
BeamRider	75.4	771	554	79.3	85.5	**885**	698	**121.7**	**474.6**	774	**718**	89.0	467.9

Table 2: Distillation results for "Student2"; **Single**: with only one teacher's supervision; **Ensemble**: proposed method with 3 teacher models; **FKL**: forward KL divergence; **RKL**: reverse KL divergence; Note: in the case of ensemble, the average of teachers' scores is used to compute the percentage.

	Single-RKL				Ensemble-FKL				Ensemble-FKL			
	Score		*Percentage*		*Score*		*Percentage*		*Score*		*Percentage*	
	Max	**Mean**	**Max**	**Mean**	**Max**	**Mean**	**Max**	**Mean**	**Max**	**Mean**	**Max**	**Mean**
Pong	13.6	12.3	101.3	100.9	**15.5**	**13.8**	**112.4**	**108.6**	15.4	13.3	109.5	104.5
Breakout	**25.5**	**15.9**	104.0	103.3	23.0	14.9	**128.2**	112.8	21.5	14.6	119.5	**113.8**
Enduro	**407**	**383**	101.8	101.7	382	313	**136.9**	**197.8**	359	301	131.2	189.9
Ms.Pacman	**1830**	**1103**	115.6	101.0	1339	1057	**126.9**	**109.8**	1299	930	120.5	97.8
Qbert	1405	1064	109.0	110.4	**4418**	**3649**	**124.2**	**119.1**	4359	3037	122.6	97.6
Seaquest	1872	1514	107.1	104.2	**1980**	**1663**	**136.1**	**133.8**	1934	1558	132.1	125.2
Riverraid	3164	2604	90.6	97.6	**4311**	**3999**	**116.4**	**110.9**	4279	3855	112.9	106.8
BeamRider	**772**	695	79.5	120.8	716	686	88.6	**467.5**	747	**712**	**91.2**	453.2

depending on the characteristics of the games. For instance, action-aware games such as Pong require harder targets by setting a smaller τ. The teachers' learning rate is the same as in DQN, but the student's learning rate is slightly higher to enable it react quickly in the distillation operation. An ϵ-greedy exploration strategy is kept with the value of 5% for all ensemble models in total. We construct three teacher models uniformly for all games, and more teacher models are suggested if the policy learning is extremely noisy.

To demonstrate the advantage in terms of distillation effectiveness and efficiency in a fair manner, we took the distillation performance of Net3 in (Rusu et al. 2015) as the benchmark. However, we emphasize that our algorithm is based on real-time policy distillation, so whether the student is able to keep up with teachers' learning makes more sense, and the mean scores of the last 10 epochs are calculated for comparison. To assess the overall performance of the ensemble method, we average the scores of the teacher models and then compute the maximum and mean scores. We also evaluate the student's distillation performance by the percentage of teacher's score or averaged teachers' scores in the ensemble method.

Results and Analysis

In Table 1 and Table 2, we present the summary of our distillation performance results, including the comparison of RKL and FKL and the comparison of a single model and ensemble models. The bold numbers show the best values among all the approaches. It is seen that the ensemble distillation is more advantageous than distillation with a single teacher model overall with respect to the percentage of maximum and mean values, and this advantage is enhanced more with RKL compared to FKL. In Table 1, Single-RKL is able to achieve competitive distillation performance as PD. The ensemble method guarantees even higher and more stable distillation performance. With respect to student's maximum and mean scores, both "Single" and "Ensemble" demonstrate similar performances. This is because if a high-performing teacher is available for the single model, the student is likely to learn a good policy during the training period. On the other hand, in the ensemble method, distillation performance might be affected by the worst teacher's supervision at first glance, but the student model still exhibits a more stable performance. Figures 3-11 show the learning curves for the ensemble distillation methods for student1 and student2. We observe that in those tasks with significant learning fluctuations, like Breakout, Ms.Pacman, and BeamRider, RKL truly helps students to resit against the uncertainty from teachers' supervision, insisting on their own opinions once they have consolidated the learned policy.

Robustness under Unstable Supervision

For better visualization of how robust ensemble policy distillation is to unstable supervision, we take two experiment runs for Seaquest, as the results shown in Figure 2. A little difference between the two implementations with respect to policy optimization. The experiment setup for the right figure is with the technique of *clip*, which sets the upper bound of gradients update. This does not fit Seaquest learning, therefore leads to performance collapse in Teacher3. The left figure shows relatively stable teachers' learning, because *clip* is removed. The comparison of the two set of plots proves that ensemble model make the distillation robust to the risk of teacher's supervision degradation. In other words, even if one teacher makes secure mistakes, its corresponding effect via supervision would be lessened. From our empirical study, a bad teacher behaves more randomly in decision making, which is reflected in its uniform-like probability distribution, providing less informative supervision to student about how to adjust parameters update comparing to stable teachers.

Conclusion and Future Work

We introduced an ensemble method for policy distillation, which guarantees relatively stable targets for the student model and makes data more diverse, thus improving the overall distillation performance. Additionally, we analyzed different properties of RKL and FKL as distillation losses and demonstrated the advantage of RKL in the Atari domain. By applying these two techniques, the compression ratio was decreased significantly with mostly full distillation on most games. Our proposed method can be applied to other scenarios, for instance, in the multi-agent training task. Our distillation method is able to assist a student in learning from multiple agents who are executing the same task but in different ways in real time, which automatically helps to mitigate negative policy. The method also probably exhibits more strength when data or experience is sparse, which is important if it is allowed only a few times to accomplish missions in the real world. One possible area of future work is making a student model learn similar tasks from multiple agents.

References

Asif, U.; Tang, J.; and Harrer, S. 2019. Ensemble knowledge distillation for learning improved and efficient networks. *arXiv:1909.08097*.

Doersch, C. 2016. Tutorial on variational autoencoders. *arXiv:1606.05908*.

Kingma, D. P., and Welling, M. 2013. Auto-encoding variational bayes. *arXiv:1312.6114*.

Kuncoro, A.; Ballesteros, M.; Kong, L.; Dyer, C.; and Smith, N. A. 2016. Distilling an ensemble of greedy dependency parsers into one mst parser. *arXiv:1609.07561*.

Lillicrap, T. P.; Hunt, J. J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; and Wierstra, D. 2015. Continuous control with deep reinforcement learning. *arXiv:1509.02971*.

Malinin, A., and Gales, M. 2019. Reverse kl-divergence training of prior networks: Improved uncertainty and adversarial robustness. *arXiv:1905.13472*.

Mirowski, P.; Pascanu, R.; Viola, F.; Soyer, H.; Ballard, A. J.; Banino, A.; Denil, M.; Goroshin, R.; Sifre, L.; Kavukcuoglu, K.; et al. 2016. Learning to navigate in complex environments. *arXiv:1611.03673*.

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.

Parisotto, E.; Ba, J. L.; and Salakhutdinov, R. 2015. Actor-mimic: Deep multitask and transfer reinforcement learning. *arXiv:1511.06342*.

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:1511.06295*.

Shen, Z.; He, Z.; Cui, W.; Yu, J.; Zheng, Y.; Zhu, C.; and Savvides, M. 2019. Adversarial-based knowledge distillation for multi-model ensemble and noisy data refinement. *arXiv:1908.08520*.

Siddique, N.; Alam, K. M. R.; and Adeli, H. 2019. A dynamic ensemble learning algorithm for neural networks. *Neural Computing and Applications*.

Sun, Y., and Fazli, P. 2019. Real-time policy distillation in deep reinforcement learning. *arXiv preprint arXiv:1912.12630*.

Zhang, X.-L., and Wang, D. 2016. A deep ensemble learning method for monaural speech separation. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)* 24(5):967–977.

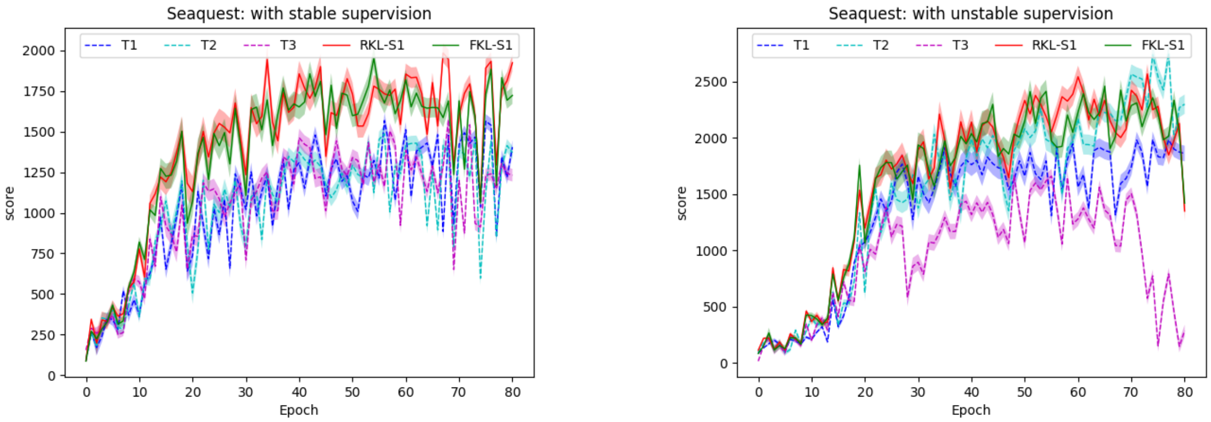

Figure 2: Comparison of distillation performance with stable vs. unstable supervision. **T1**: Teacher1, **T2**: Teacher2, **T3**: Teacher3, **RKL**: reverse KL divergence, **FKL**: forward KL divergence.

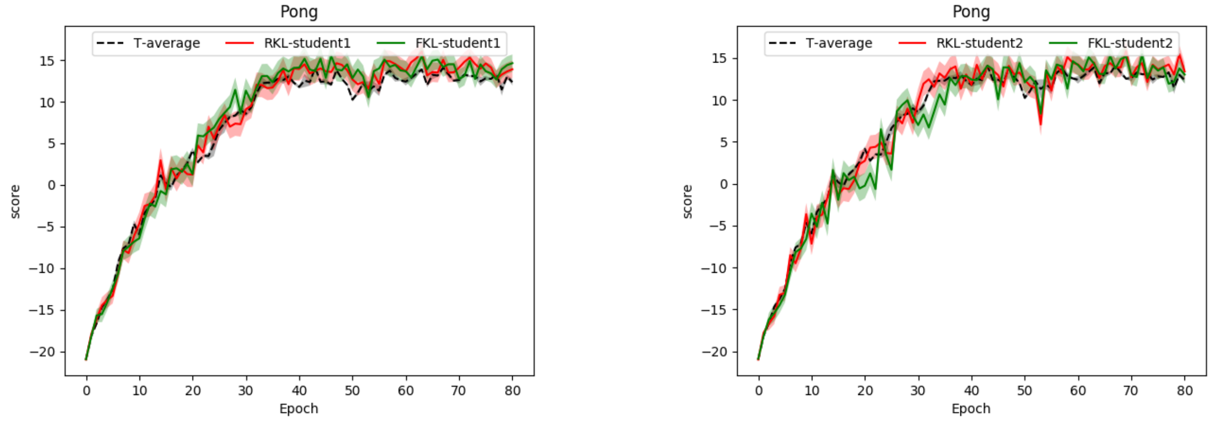

Figure 3: Learning Curves of **Pong**. *Left figure*: student1; *Right figure*: student2. **T-average**: Teachers average score, **RKL**: reverse KL divergence, **FKL**: forward KL divergence.

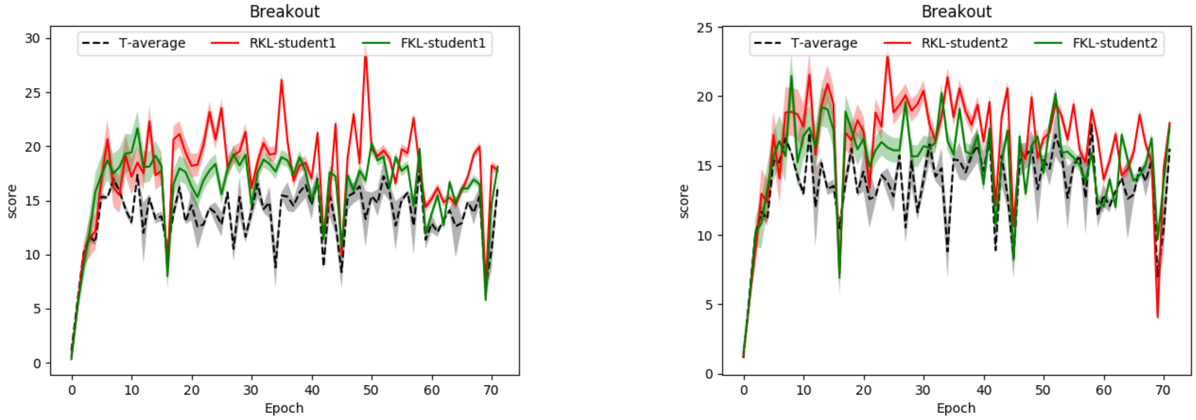

Figure 4: Learning Curves of **Breakout**. *Left figure*: student1; *Right figure*: student2. **T-average**: Teachers average score, **RKL**: reverse KL divergence, **FKL**: forward KL divergence.

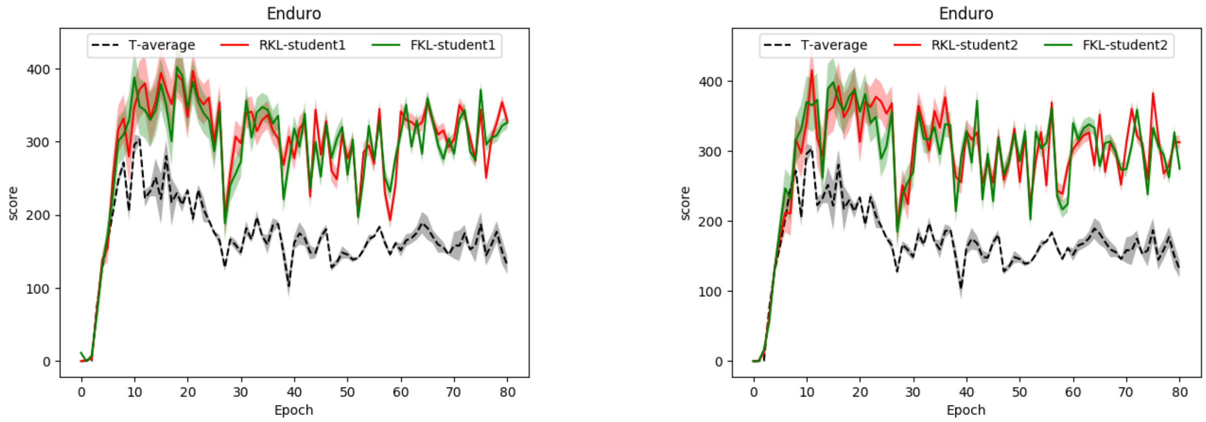

Figure 5: Learning Curves of **Enduro**. *Left figure*: student1; *Right figure*: student2. **T-average**: Teachers average score, **RKL**: reverse KL divergence, **FKL**: forward KL divergence.

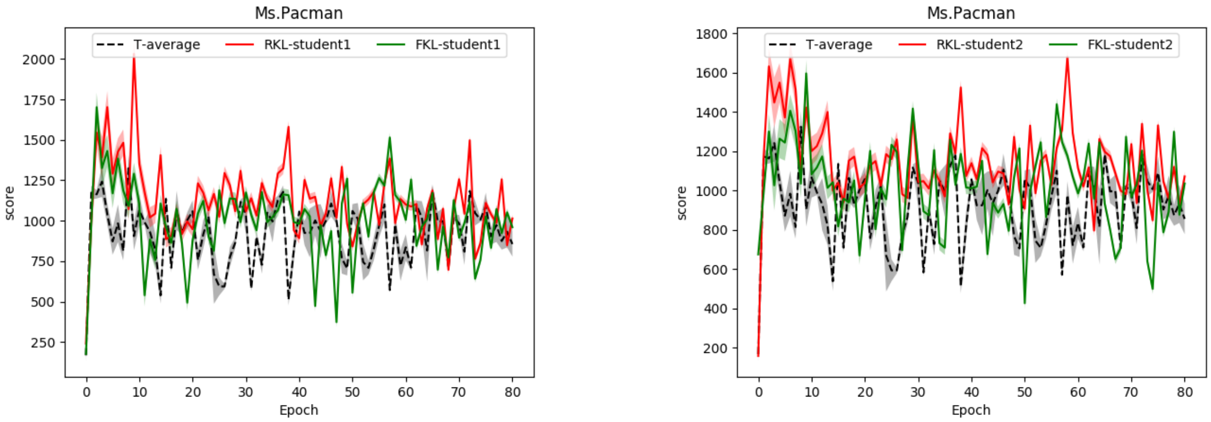

Figure 6: Learning Curves of **Qbert**. *Left figure*: student1; *Right figure*: student2. **T-average**: Teachers average score, **RKL**: reverse KL divergence, **FKL**: forward KL divergence.

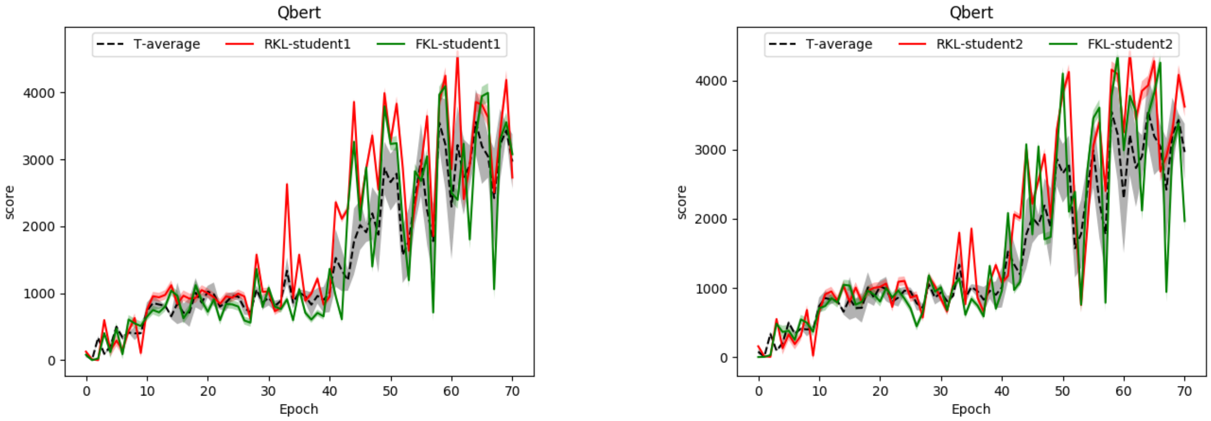

Figure 7: Learning Curves of **Qbert**. *Left figure*: student1; *Right figure*: student2. **T-average**: Teachers average score, **RKL**: reverse KL divergence, **FKL**: forward KL divergence.

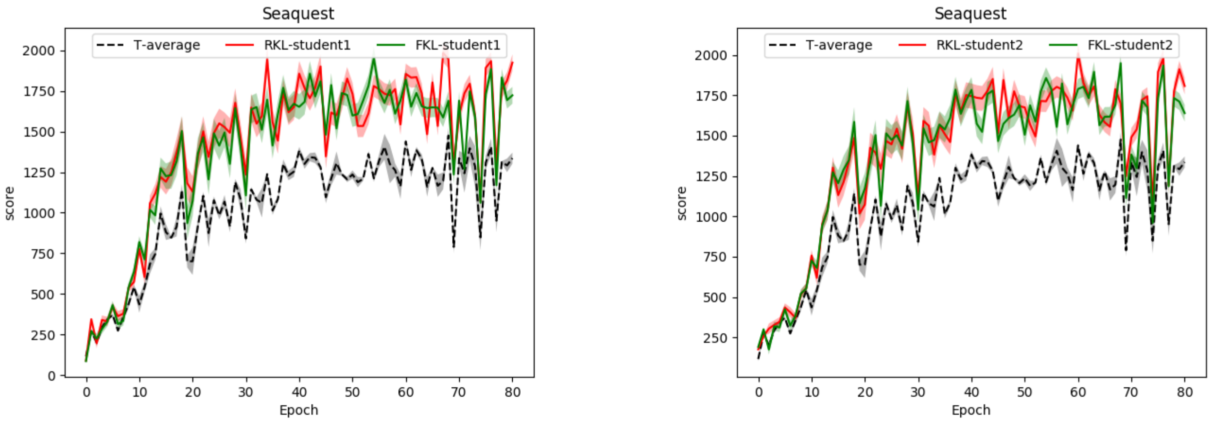

Figure 8: Learning Curves of **Seaquest**. *Left figure*: student1; *Right figure*: student2. **T-average**: Teachers average score, **RKL**: reverse KL divergence, **FKL**: forward KL divergence.

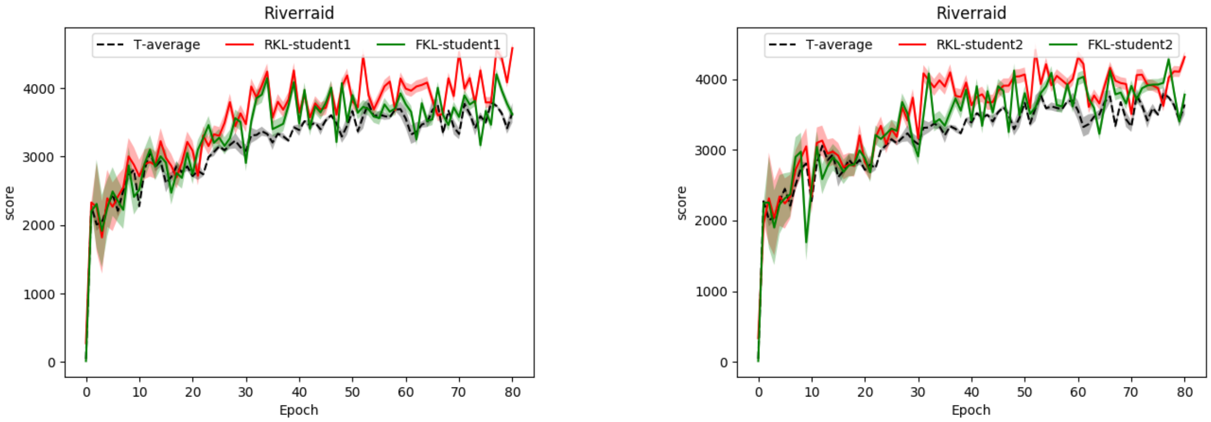

Figure 9: Learning Curves of **Riverraid**. *Left*: Student1; *Right*: Student2. **T1**: Teacher1, **T2**: Teacher2, **T3**: Teacher3, **S1**: Student1, **S2**: Student2, **RKL**: reverse KL divergence, **FKL**: forward KL divergence.

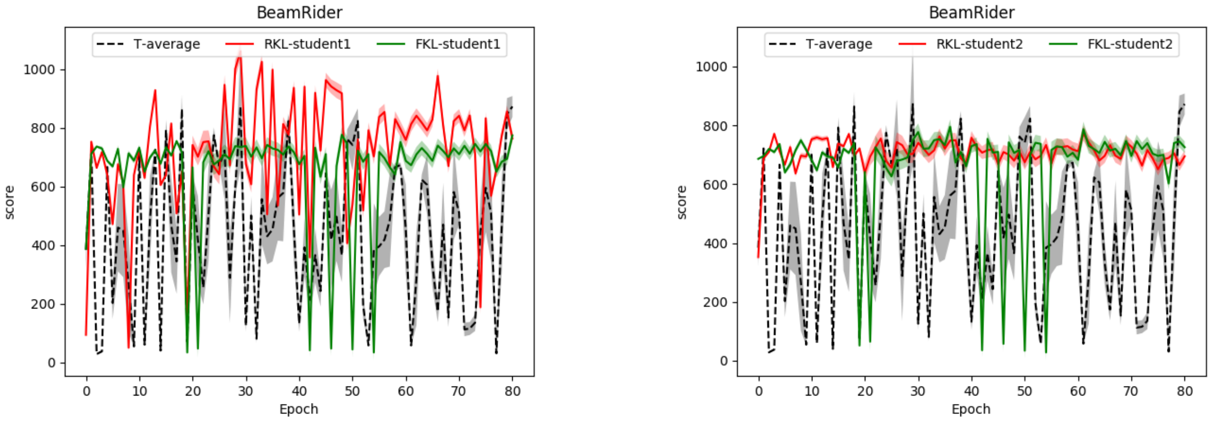

Figure 10: Learning Curves of **BeamRider**. *Left figure*: student1; *Right figure*: student2. **T-average**: Teachers average score, **RKL**: reverse KL divergence, **FKL**: forward KL divergence.

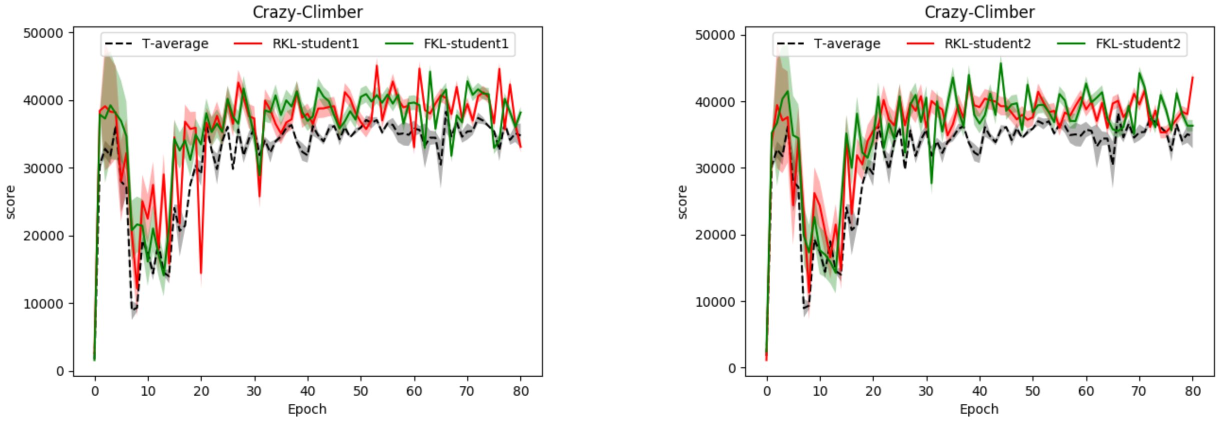

Figure 11: Learning Curves of **Crazy-Climber**. *Left figure*: student1; *Right figure*: student2. **T-average**: Teachers average score, **RKL**: reverse KL divergence, **FKL**: forward KL divergence.