



Approaches to Policy Advice From Multiple Teachers in Reinforcement Learning

Link

<http://nrs.harvard.edu/urn-3:HUL.InstRepos:38811482>

Terms of use

This article was downloaded from Harvard University's DASH repository, and is made available under the terms and conditions applicable to Other Posted Material (LAA), as set forth at

<https://harvardwiki.atlassian.net/wiki/external/NGY5NDE4ZjgzNTc5NDQzMGIzZWZhMGFIOWI2M2EwYTg>

Accessibility

<https://accessibility.huit.harvard.edu/digital-accessibility-policy>

Share Your Story

The Harvard community has made this article openly available.
Please share how this access benefits you. [Submit a story](#)

Acknowledgements

First and foremost, I would like to thank Ofra Amir for advising me throughout the research process, without whom I would be playing Pac-Man rather than teaching agents to play it. I would also like to give my sincere thanks to Professor Barbara Grosz for being my advisor during the research process and giving me the initial inspiration for this project. I also thank Professor Scott Kuindersma for generously lending his time to be a reader. Finally, I would also like to thank my friends and family for their undying support of my work.

Contents

Abstract	i
Acknowledgements	ii
List of Figures	v
List of Tables	vii
1 Introduction	1
1.1 Overview	1
1.2 Motivation	2
2 Related Work	4
3 Background	5
3.1 Reinforcement Learning	5
3.1.1 Q-Learning	6
3.1.2 SARSA Learning	7
3.1.3 Eligibility Trace	7
3.2 The Student-Teacher Framework	7
3.3 Multi-Teacher Framework	8
4 Methods	9
4.1 Domain	9
4.2 Experimental Settings	10
4.3 Teaching Heuristics	11
4.3.1 Early Advising	11
4.3.2 Importance Advising	11
4.3.3 Mistake Correcting	12
4.4 Multi-Teacher Aggregation	13
4.4.1 Grand-Teacher	13
4.4.2 Individual Teachers	14
5 Results	16
5.1 Single Teacher Heuristics	16
5.2 Multi-Teacher Heuristics	17
5.2.1 Choosing When to Advise - Majority Vote	18

5.2.1.1	Similar Teachers	18
5.2.1.2	Similar Teachers Revisited	20
5.2.1.3	Diverse Teachers	20
5.2.1.4	Quantifying Budget Usage	23
5.2.1.5	Budget Usage - Similar Teachers (Reversed)	24
5.2.1.6	Budget Usage - Diverse Teachers	25
5.2.2	Choosing When to Advise - Independent Teachers	30
5.2.2.1	Similar Teachers	37
5.2.2.2	Similar Teachers Revisited	37
5.2.2.3	Diverse Teachers	38
5.2.2.4	Budget Usage - Similar Teachers (Reversed)	39
5.2.2.5	Budget Usage - Diverse Teachers	42
5.2.3	Comparing Majority Vote with Independent Teachers	42
6	Conclusion and Future Work	48
	Bibliography	49

List of Figures

4.1	Pac-Man Gameplay	10
5.1	Single Teacher Importance Advising Thresholds	17
5.2	Single Teacher Mistake Correcting Thresholds	17
5.3	Single Teacher vs. No Teacher	18
5.4	Similar Teachers (Early Advising)	19
5.5	Similar Teachers (Importance Advising)	19
5.6	Similar Teachers (Mistake Correcting)	20
5.7	Similar Teachers Reversed (Early Advising)	21
5.8	Similar Teachers Reversed (Importance Advising)	21
5.9	Similar Teachers Reversed (Mistake Correcting)	21
5.10	Diverse Teachers (Early Advising)	23
5.11	Diverse Teachers (Importance Advising)	23
5.12	Diverse Teachers (Mistake Correcting)	23
5.13	Budget Usage of 1 Teacher (Reversed)	25
5.14	Budget Usage of 2 Similar Teachers (Reversed)	26
5.15	Budget Usage of 3 Similar Teachers (Reversed)	27
5.16	Budget Usage of 4 Similar Teachers (Reversed)	28
5.17	Budget Usage of 5 Similar Teachers (Reversed)	29
5.18	Budget Usage of 1 Teacher (T=400)	30
5.19	Budget Usage of 2 Diverse Teachers (T=400, 500)	31
5.20	Budget Usage of 2 Diverse Teachers (T=400, 1000)	32
5.21	Budget Usage of 2 Diverse Teachers (T=600, 1000)	33
5.22	Budget Usage of 3 Diverse Teachers (T=400, 500, 1000)	34
5.23	Budget Usage of 3 Diverse Teachers (T=400, 600, 1000)	35
5.24	Budget Usage of 4 Diverse Teachers (T=400, 500, 600 1000)	36
5.25	Independent Similar Teachers (Importance Advising)	37
5.26	Independent Similar Teachers (Mistake Correcting)	37
5.27	Independent Similar Teachers Reversed (Importance Advising)	38
5.28	Independent Similar Teachers Reversed (Mistake Correcting)	38
5.29	Independent Diverse Teachers (Importance Advising)	39
5.30	Independent Diverse Teachers (Mistake Correcting)	39
5.31	Budget Usage of 2 Independent Similar Teachers (Reversed)	40
5.32	Budget Usage of 3 Independent Similar Teachers (Reversed)	40
5.33	Budget Usage of 4 Independent Similar Teachers (Reversed)	41
5.34	Budget Usage of 5 Independent Similar Teachers (Reversed)	41
5.35	Budget Usage of 2 Independent Diverse Teachers (T=400, 500)	42
5.36	Budget Usage of 2 Independent Diverse Teachers (T=400, 1000)	43

5.37 Budget Usage of 2 Independent Diverse Teachers (T=600, 1000)	43
5.38 Budget Usage of 3 Independent Diverse Teachers (T=400, 500, 1000)	44
5.39 Budget Usage of 3 Independent Diverse Teachers (T=400, 600, 1000)	44
5.40 Budget Usage of 4 Independent Diverse Teachers (T=400, 500, 600 1000)	45

List of Algorithms

1	Early Advising	11
2	Importance Advising	12
3	Mistake Correcting	13
4	Online Grand-Teacher	14
5	Online Independent Teachers	15

List of Tables

5.1	Significance Comparison: Independent Teachers more effective than Majority Voting (Similar Teachers)	45
5.2	Significance Comparison: Independent Teachers more effective than Majority Voting (Diverse Teachers)	46
5.3	Significance Comparison: Majority Voting more effective than Independent Teachers (Similar Teachers)	46
5.4	Significance Comparison: Majority Voting more effective than Independent Teachers (Diverse Teachers)	47

Chapter 1

Introduction

Learning a task can be tediously slow or repetitive. Through schooling, teachers can help ease the process of learning, helping students learn faster and develop the potential to learn beyond what they could have learned individually. Similar to human teachers, software agents can decrease learning time as teachers for human students or student agents.

1.1 Overview

In reinforcement learning (RL), independent agents learn how to perform a sequential decision process. This usually requires a lone agent to repeatedly complete training episodes and learn through trial-and-error. Thus, RL is commonly understood to be computationally expensive and slow.

This work focuses on how RL agents can act as teachers and help other RL agents master a task more quickly, potentially saving resources and time. This work implements an RL agent as the student, but it uses a framework that can potentially expand to human students. More specifically, in order to generalize to human students, I focus on a human-understandable teaching method, do not assume teachers know a student’s internal policy, and do not simply infuse the student with the teacher’s policy. In this case, I use the action advice model [1]. During the student’s training, the teacher observes the student’s actions (without knowledge of the student’s policy) and suggests actions to take (a method understandable by a human aware of the action space). The action advice model only assumes the teacher and student have the same action space (the set of all possible actions to take), which is conducive not only to extending to human students but also to connecting agents with possibly different state spaces (the set of all the states of an environment).

However, this action advice model also requires the teachers' continual attention to the student, as the student constantly requires the teacher's advice or at least the teacher's policy for all possible states. Therefore, this work limits each teacher with an advice budget as suggested by Torrey and Taylor [1]; a teacher can only give so much advice and thus must strategically choose when to dispense advice to help the student the most. Similar to Torrey and Taylor [1], I establish the budgets for several reasons. It is not reasonable for a teacher to give advice at every step in the student's learning, either for the student's or for the teachers' resources. Some domains also may not allow the teachers to connect with the student at every step of the learning, such as if the student needs more than one time step to process advice. In addition, over-advising can also mislead or hamper the student's learning, especially for teachers with state spaces different from the student's. This is also important if the model is extended to human teachers, as humans may have limited attention or patience, though this is not explored in this thesis.

In addition, I also focus on the effect of multiple teachers teaching one student. If one teacher is beneficial to accelerating a student's learning, how beneficial are two or more teachers? If multiple teachers give different advice, is it possible for a student to benefit more than if it only had one of the teachers? Zhan et al. established a multi-teacher advice model [2], and I implement their approach as well as propose a modification in the algorithm in an empirical analysis of the performance of multiple teachers in the Pac-Man domain. Results show that the multi-teacher framework can outperform single teachers, even with budgeted teachers. I also discover that the choice of the teachers is very important for the student's learning.

1.2 Motivation

The purpose of introducing teachers is to reduce learning time and possibly increase maximum skill level. This can be applied to any RL problem, but a handful of motivating examples are included in this section.

One example is on-boarding for a new computer program. When introduced with a program with unfamiliar functionality, whether it is a text editor, a computational tool, or even music-producing software, a new user may face difficulty learning how to use it. If a teacher agent was able to guide the users while they learned, learning time could be reduced and productivity with the program can be increased, thus optimizing the usefulness of the program.

Another example is teaching a robot to navigate an obstacle, such as climbing stairs. This student is now an agent that could benefit from one or more teacher agents advising how it can maneuver its individual parts to complete the task, allowing it to learn faster and become more optimal.

Yet another example is learning how to play a video game. If a new player is learning how to play, a teacher agent can help guide the player and teach a strategy to orient the player more quickly. Furthermore, multiple teachers with possibly differing strategies can simultaneously advise the player. Besides making the game easier to learn and thus more enjoyable, teachers could help reduce developer resources required to implement a tutorial functionality for new players.

This work focuses on this application of teachers in RL, using teacher agents to teach student agents how to play the Pac-Man video game. Though I focus solely on student agents and teacher agents, I construct the framework to be constrained as in Torrey and Taylor's work [1] such that it may be extended to human students.

Chapter 2

Related Work

This research extends previous work on the student-teacher reinforcement learning framework as well as studies on the multi-teacher reinforcement learning framework [2].

Torrey and Taylor [1, 3] explored different algorithms for a teacher on a limited budget. More specifically, they looked into Early Advising, Importance Advising, Mistake Correcting, Alternate Advising, and Predictive Advising for the student-teacher reinforcement learning framework when the teacher has a limited budget (the details of some of these heuristics are outlined in [chapter 3](#)). Zhan et al. [4] similarly used these same heuristics but purely in the context of an agent teaching a human student. This thesis borrows several teacher heuristics from their methods but applies it to a multi-teacher framework.

Zhan and Taylor [2] investigated several multi-teacher algorithms, including the Grand-Teacher model, which treats multiple teachers as one, advising the majority action of the teachers. However, their approach did not investigate budget sizes smaller than the state space and thus does not require budgeted advising heuristics. This thesis extends their Grand-Teacher model but imposes a stricter budget on all the teachers and investigates different implementations that do not need to traverse all the states, namely online algorithms.

This research differs from each of these related studies individually; I borrow ideas from each to investigate a possibly new multi-teacher student-teacher reinforcement learning framework. Specifically, I apply Torrey, Taylor, and Zhan’s [1, 4] budgeted teacher algorithms to multiple teachers in ways reminiscent of Zhan and Taylor’s [2] Grand-Teacher model and empirically observe the results of the different algorithms in the Pac-Man domain.

Chapter 3

Background

This section gives an introductory background to reinforcement learning, including the learning algorithms used in this research, as well as the framework used to incorporate one or more teachers in reinforcement learning.

3.1 Reinforcement Learning

Reinforcement learning (RL) is a machine learning method for an agent to learn how to behave in some environment, specifically in a Markov decision process (MDP). A decision process is a framework in which an agent sequentially performs actions and receives different rewards depending on those actions. A decision process is Markovian if its future is conditionally independent of the past given the present. That is, an agent in an MDP lives “in the moment” such that its current state and action decide its fate regardless of its previous actions.

In RL, an agent is aware of the state space S , the set of all the states in the world. The agent also knows the action space A , the set of all possible actions it can take, as well as the action model, the set of possible actions at a particular state. In an MDP, the agent can observe the reward from taking a particular action $a \in A$ at a certain state $s \in S$ to move to a new state $s' \in S$ and thus acts to maximize its reward. However, in RL, the agent does not know the transition model $p(s'|s, a)$, the probabilistic model that describes what states result from any state/action pair. The agent also does not know the reward function R , the model that denotes what reward results from any state/action pair, but an RL agent can observe these. In this way, an RL agent learns to maximize its reward through trial-and-error, usually through many trials, or episodes.

During this process, an RL agent maintains its policy $\pi(s)$, which is its guide on what action to perform at a state s in order to maximize its reward. At every step, when the agent observes its

current state s , takes action a , and observes resulting reward r and transition state s' , the agent updates its policy, iterating until it has completed all of its independent training episodes.

In model-based learning, an RL agent can learn what decisions to make by estimating the transition model and the reward function. However, in practice, this is not always achievable; an agent needs a very large number of samples to get a reasonable estimate of the model. Thus, in model-free learning, an agent directly estimates the optimal policy, without having to construct the entire MDP model. Two common methods for learning an optimal policy are Q-Learning and SARSA Learning.

3.1.1 Q-Learning

In Q-Learning, an agent represents its policy with a Q-Function, $Q(s, a)$, which estimates the reward resulting from performing action a in state s . Thus, at a fixed state s , a Q-Learning agent would want to take the action a maximizing $Q(s, a)$. Exploration updates the Q-Function. However, RL agents may learn to repeat the same actions that maximize their current Q-Function. To encourage exploration and avoid any local maxima in an agents' reward that could prevent it from exploring a better policy, the agent has a probability ϵ that it performs a random action instead of relying on maximizing its Q-Function. This type of exploration is called ϵ -greedy.

Q-Functions can be represented as direct mappings of state/action pairs with Q-values, but in practice, the environment is usually too large and complex to reasonably maintain a simple table for the Q-Function. Thus, states can be approximated with a set of features $\{f_1, f_2, \dots, f_n\}$. Such features may categorize the states to simplify the environment, or they may be useful indicators of particular states. Thus, a Q-learning agent maintains a set of weights $\{w_1, w_2, \dots, w_n\}$ and updates it to update its Q-Function, represented as follows:

$$Q(s, a) = \sum_i^n w_i f_i$$

In general, in Q-Learning, an agent updates its Q-Function when it performs action a at state s , observing a reward of r and a transition state of s' in the following formula:

$$Q(s, a) \leftarrow Q(s, a) + \alpha \left[(r + \gamma \max_{a' \in A} Q(s', a')) - Q(s, a) \right]$$

where $0 < \alpha < 1$ is the learning rate, how much new information will update old information, and $0 < \gamma < 1$ is the discount factor, how much the agent prioritizes long-term reward. Thus, ϵ -greedy Q-Learning takes in three parameters, $\{\epsilon, \alpha, \gamma\}$.

In addition, the initial values for the weights can be optimistic, such that they start high and are adjusted lower, or pessimistic, such that they start low and are adjusted higher. Agents with optimistic values tend to explore more, as they have a more optimistic view of the unexplored. Conversely, agents with pessimistic values tend to rely on repeating successful actions, so if a teacher were to advise an agent on what actions to take, an agent with pessimistic values would benefit more from the advice by relying more heavily on what it learns through past advice than would an agent with optimistic values, so this research focuses only on agents with pessimistic starting weight values.

3.1.2 SARSA Learning

SARSA (State-Action-Reward-State-Action) is a learning algorithm very similar to Q-Learning, but a SARSA agent updates its Q-Function slightly differently. Instead of greedily choosing the action with the greatest Q-value, SARSA will calculate the resulting Q-value from the current policy's action:

$$Q(s, a) \leftarrow Q(s, a) + \alpha [(r + \gamma Q(s', a')) - Q(s, a)]$$

where α and γ are identical to the ones in the Q-Learning formula, but a' is now the policy's action for state s' . While Q-Learning does not depend on the policy and is thus an off-policy algorithm, SARSA is on-policy because it requires knowledge of a' . Thus, SARSA treats exploration in a more realistic setting, considering actions that would follow policy.

3.1.3 Eligibility Trace

The eligibility trace (λ) can be used with Q-Learning or SARSA in well-established algorithms in RL. This eligibility trace provides a temporary record of a state or action to better inform learning in training [5]. This paper uses the same $Q(\lambda)$ and $SARSA(\lambda)$ with linear Q-Function approximation used in previous works [1, 3] on the student-teacher framework.

3.2 The Student-Teacher Framework

The Student-Teacher RL framework, as introduced by Torrey and Taylor [1], naturally includes two agents: the student and the teacher. The teacher, presumably with more learning experience or better insight, has its own fixed policy, while the student uses an RL algorithm, such as Q-Learning or SARSA learning, to learn a task. In action advice learning, at certain states during the student's learning, the teacher advises an action for that state, and the student takes that action, treating it as a regular action and updating its Q-Function according to its learning algorithm as normal.

In budgeted teaching, the teacher has a set limit on the number of actions it can advise. This is a more reasonable and realistic narrowing of the framework, as teachers cannot always be expected to have the resources to readily advise every step of the student’s learning.

3.3 Multi-Teacher Framework

The Multi-Teacher framework, as introduced by Zhan et al. [2], extends the Student-Teacher framework in that two or more teachers advise one student. In this sense, each teacher has its own policy and budget. However, questions arise as to how a single student can reconcile potentially different advice received from multiple sources simultaneously.

Several algorithms exist for aggregating multiple teachers for policy advice, including the Grand-Teacher procedure [2]. Zhan et al. also introduce an advice algorithm that combines the Grand-Teacher model with an algorithm introduced by Bartlett and Tewari [6], REGAL.C, which the student uses to estimate the MDP in order to learn an optimal policy (model-based learning) [2].

However, this thesis focuses on the online Grand-Teacher model and model-free learning for the student. The reasoning for this is these algorithms are usually simpler and more intuitive, and they are more comparable to each other than to more complex algorithms. The Q-Learning and SARSA learning used for the student also avoids having to construct the entire MDP, which would be computationally expensive for the Pac-Man domain as it has a very large state space.

Chapter 4

Methods

This section clarifies the methods used to demonstrate the teaching strategies and multi-teacher framework I explored in application to the Pac-Man domain. After an introduction of the Pac-Man domain, this section will detail the experimental settings and explain the teaching heuristics used in this research (Early Advising, Importance Advising, and Mistake Correcting) and the different implementations of multi-teacher aggregation.

4.1 Domain

This study uses the Ms. Pac-Man video game domain (see Figure 4.1), as used by Taylor et al. [3]. Ms. Pac-Man was a classic arcade game that was part of the Pac-Man series, a game where an agent navigates a two-dimensional maze collecting food and avoiding four ghosts (or defeating the ghosts with a powerup pickup, a larger food pellet that temporarily allows the ghosts to be defeated if Pac-Man touches them).

An episode consists of a contained game in the same maze, a life of the Pac-Man agent; in other words, each episode begins the same way and ends if one of the following occurs: Pac-Man is defeated by a ghost, Pac-Man eats all the food in the maze, or Pac-Man has reached 2000 steps without winning or being defeated. Each episode begins the same way, with Pac-Man at its starting point and the four ghosts together in their starting places. The ghosts move around randomly, but Pac-Man can choose to move in any of the four cardinal directions (NORTH, EAST, SOUTH, or WEST) at each step, provided there is space to move in that direction.

Pac-Man’s performance is measured with a score system. Each food pellet Pac-Man picks up is 10 points, and each powerup pickup is 50 points and also allows Pac-Man to defeat a ghost for 200 points (thus causing the ghost to respawn in its starting place) in a limited timeframe.

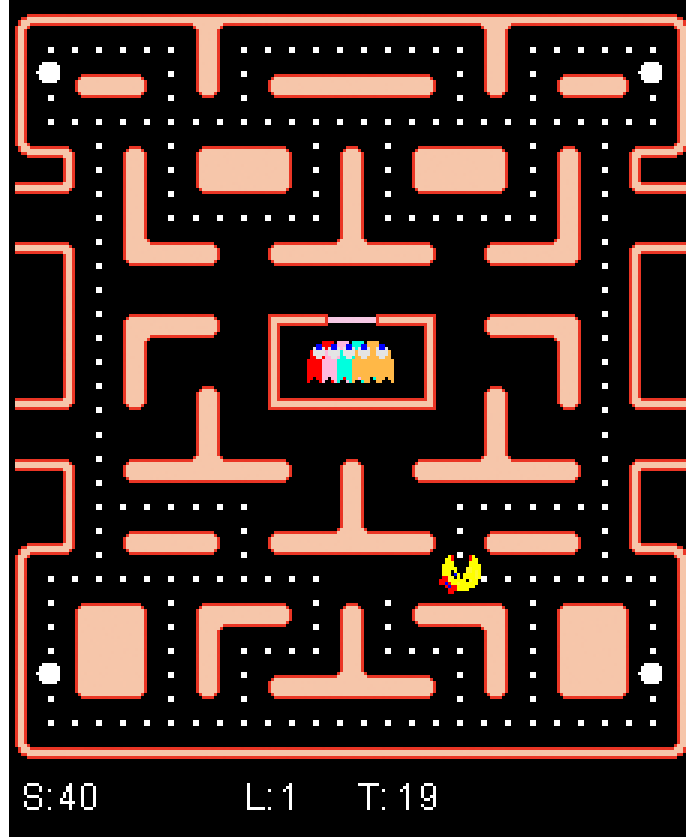


FIGURE 4.1: Screenshot of Ms. Pac-Man domain

Each state in this domain consists of Pac-Man’s position, the ghosts’ positions, and the positions of all the food pellets and powerup pickups. Similar to Taylor et al. [3], this research utilizes Q-function approximation with feature sets, due to the very high number of distinct states. The features are identical to Taylor’s “low-asymptote” set [3] to allow for easier comparison of results, and the features include metrics such as distance from Pac-Man to food pellets, and distance to ghosts.

4.2 Experimental Settings

The RL parameters used in this study were $\epsilon = 0.05$, $\alpha = 0.001$, $\gamma = 0.999$, and $\lambda = 0.9$. These are kept identical to Taylor and Torrey’s parameters [1, 3] for consistency. I implement both $Q(\lambda)$ and $SARSA(\lambda)$ students.

For each teacher, I also establish a budget of size $n = 1000$, which is half the number of steps in a game played through without ending prematurely.

A student’s progress is recorded by pausing training every 10 episodes to perform 30 evaluation episodes of the student alone. In these evaluation episodes, no additional learning, teaching, or exploration occurs. This entire process is repeated 30 times and averaged to obtain the student’s

learning curve. The performance of a student is measured by the area under its learning curve; this shows if the student scores higher on average earlier during its training (steeper slope) or whether it scores higher on average (greater asymptote), both of which are desirable in RL. All the parameters, state representations, and individual learning algorithms were kept constant so that any difference in the learning curve resulted from teaching.

I built on Taylor et al.’s Java code framework [3]. My code is publicly available on my Github: <https://github.com/tonyli1/Multi-Teacher-RL-PacMan>.

4.3 Teaching Heuristics

In this research, I focus on three of the teaching heuristics explored by Torrey and Taylor for teachers on budgets: Early Advising, Importance Advising, and Mistake Correcting [1].

4.3.1 Early Advising

In Early Advising, a teacher gives advice to the student for the first b states the student encounters, if b is the teacher’s budget. The idea is for the teacher to give the student a strong, advised start at the beginning so that it can learn earlier in its exploration. The algorithm is outlined in Algorithm 1. This is used as the baseline.

Algorithm 1: Early Advising

Given teacher policy π , teacher’s remaining budget n ;

for each state s **do**

if $n > 0$ **then**

$n = n - 1$;

 Advise($\pi(s)$);

end

end

4.3.2 Importance Advising

For teachers on budgets, Early Advising quickly uses up their budgets. In a sequential decision process scenario like Pac-Man, we’d expect some states to be more important than others; certain actions in some states can have more extreme rewards or punishments than actions in other states. Thus, Importance Advising allows teachers to dispense advice in more important states, to save its budget for more crucial points in the students’ learning.

With Q-function learning, a threshold can naturally be drawn to determine state importance. Because a teacher has a particular Q-value for each state and action pair ($Q(s, a)$), importance can be interpreted as the difference between the maximum and minimum Q-value for actions in a certain state:

$$I(s) = \max_a(Q(s, a)) - \min_a(Q(s, a))$$

If this importance is above a set threshold, then the teacher decides to dispense advice to the student. One wrong move could cause the student's performance to plummet, while making another move can increase performance and learning. In the case of Pac-Man, an example of an important state is as follows: if Pac-Man is in a state such that moving WEST would result in being defeated by a ghost but moving EAST would result in Pac-Man eating a powerup pickup, then this is likely an important state because the Q-Values resulting from this state range from that of losing the game to that of eating a powerup pickup and being able to defeat ghosts. Notice that Early Advising is equivalent to Importance Advising with the threshold set to 0. The algorithm is outlined in Algorithm 2.

Algorithm 2: Importance Advising

Given teacher policy π , teacher budget n , threshold t ;

for each state s **do**

if $n > 0$ and $I(s) \geq t$ **then**

$n = n - 1$;

 Advise($\pi(s)$);

end

end

4.3.3 Mistake Correcting

For Importance Advising, a teacher can waste its budget if it advises an action the student was already planning to take. Similarly to Importance Advising, Mistake Correcting checks for state importance, but a teacher following this heuristic only advises the student if the student's intended announced action differs from the action the teacher would advise with its own policy.

This algorithm is outlined in Algorithm 3.

Algorithm 3: Mistake Correcting

Given teacher policy π , teacher budget n , threshold t ;

for each state s **do**

a = student's announced action;

if $n > 0$ and $I(s) \geq t$ and $a \neq \pi(s)$ **then**

$n = n - 1$;

 Advise($\pi(s)$);

end

end

4.4 Multi-Teacher Aggregation

At each step, a student can receive advice from a teacher if the teacher has not used up its budget and if the teacher chooses to advise. However, this study explores the case of multiple teachers advising a single student. At a step, a student might have to process advice from several teachers with different strategies or otherwise combine the teachers into a single entity to generalize to the single-teacher model. Several algorithms exist for multi-teacher aggregation, but I focus on the Grand-Teacher model and investigate my changes to that model, in the Independent Teachers algorithm.

4.4.1 Grand-Teacher

One way to aggregate multiple teachers is to combine them into one grand-teacher, with a unified policy. The grand-teacher can be constructed online or offline.

In the online grand-teacher model, the teachers process each new state along with the student as the student reaches it, sending their advice based on their individual policies $\pi_i \in \Pi$. The student then chooses the majority action (or randomly chooses from the most-advised actions if there is a tie) and treats it as it does in the single teacher model. The online grand-teacher thus requires all the teachers to be available at every step it provides advice.

In the offline grand-teacher model, the teachers traverse every state in the MDP and store the majority action in a meta-policy. Unlike the online implementation, this does not require the continuous availability of the teachers during the student's learning, but it requires extra traversal of the entire MDP, which can be intractable for large state spaces. In these cases, an approximation of the state space can be made with an upper limit on the number of steps [2].

I only focus on the online construction of the grand-teacher in this work, because it is simpler and does not require a traversal of the entire MDP. A basic algorithm of the online Grand-Teacher is outlined in Algorithm 4.

Algorithm 4: Online Grand-Teacher

Given teacher policies Π ;

for each state s **do**

if $ShouldAdvise(s)$ **then**

 Advise(MajorityVote($\Pi(s)$));

end

end

Zhan et al. assumed each teacher’s budget is equal to the size of the state space ($b_1 = \dots = b_m = |S|$) [2]. In this thesis, however, I impose a set budget ($b_1 = \dots = b_m = 1000$) and thus implement the Early Advising, Importance Advising, and Mistake Correcting heuristics with multiple teachers.

To simplify the implementation, for this majority-voting Grand-Teacher model, I established one fixed budget b for the budget of all the teacher’s advising, so that all the teachers either advise or don’t advise, together. I made this decision because it would make sense in many domains for the teachers to be restricted in how many times they can interact with the student. Thus, the advising heuristics for multiple teachers extend naturally from the single-teacher model. When the teachers check importance (i.e. Importance Advising and Mistake Correcting), each teacher checks the importance of the student’s state based on their individual policies, and if a majority of them deem the state important based on their own thresholds, the teachers then advise their majority action to the student.

4.4.2 Individual Teachers

An algorithm this paper introduces for this multi-teacher framework is a model that leverages each teacher individually. Instead of a majority vote to decide when all the teachers (aggregated as the Grand-Teacher) choose to advise the student in Importance Advising and Mistake Correcting, each teacher individually chooses when to advise the student, so that at any state, the student could receive advice from 0 to T teachers, if there are T total teachers. The student always chooses the majority-advised action at any state.

In this thesis, each teacher maintains its own budget equal in size to the total budget used in the Grand-Teacher model. The reasoning for this was to make sure each teacher’s voice is heard; every teacher has the chance to input their own “specialty” and fill in advice even when the other teachers have used up their budget or do not have as much experience in certain states.

This may not be a reasonable setup for domains that limit interaction with the student, but it still imposes a restriction on the teachers' side. Revising this algorithm to use an overall budget (such that the budget is used in each step when any teacher advises) would make this applicable to the same domains to which the Grand-Teacher can be applied. An algorithm for an online interpretation is outlined in Algorithm 5.

Algorithm 5: Online Independent Teachers

Given teacher policies Π ;

for *each state* s **do**

 Initialize Π^* empty list of policies;

for *each teacher* t **do**

if t *ShouldAdvise*(s) **then**

 Add policy π_t to Π^* ;

end

end

 Advise(MajorityVote($\Pi^*(s)$))

end

Chapter 5

Results

The first part of this section presents results from a single teacher using several teaching heuristics on Taylor’s low-asymptote feature sets [3]. This was mainly to replicate Taylor and Torrey’s results [1, 3] and demonstrate the effect of a teacher in a student-teacher framework.

The rest of this section presents the results from exploring the multi-teacher framework with different aggregation heuristics and with different types of students and teachers. Results are shown for both the majority-voting Grand-Teacher framework and the Independent Teachers framework.

5.1 Single Teacher Heuristics

I began with replicating Taylor’s results with a single teacher in the student-teacher framework. I first trained 30 independent students each for 1000 training episodes (because based on Taylor’s results of the single-teacher heuristics, this appears to be sufficient for a student to reach the asymptotic maximum of its skill in Pac-Man), and I chose the policy of the top-performing teacher (when evaluated against the other teachers) for the teacher in this framework.

I compared different importance thresholds for this single teacher in Importance Advising and Mistake Correcting, and I found that a threshold of 40 appeared to be most effective for teaching for both Q-learning and SARSA-learning students, as visualized in Figures 5.1 and 5.2. Thus, I used this threshold for both heuristics.

I trained every new student agent for 1000 episodes, evaluating every 10 episodes and averaging over 30 runs, first without the teacher then using the Early Advising, Importance Advising, and Mistake Correcting heuristics with the teacher. I completed all the testing with $Q(\lambda)$ and $SARSA(\lambda)$ students.

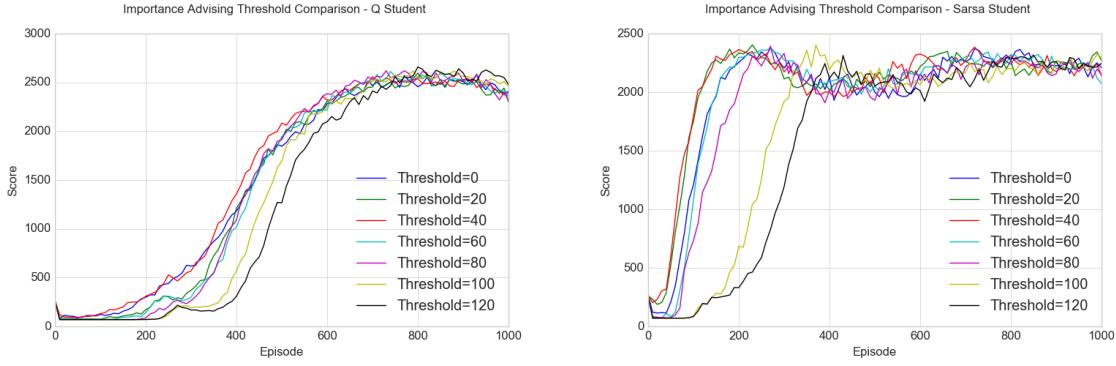


FIGURE 5.1: Threshold comparison for Importance Advising, for Q-student and SARSA-student

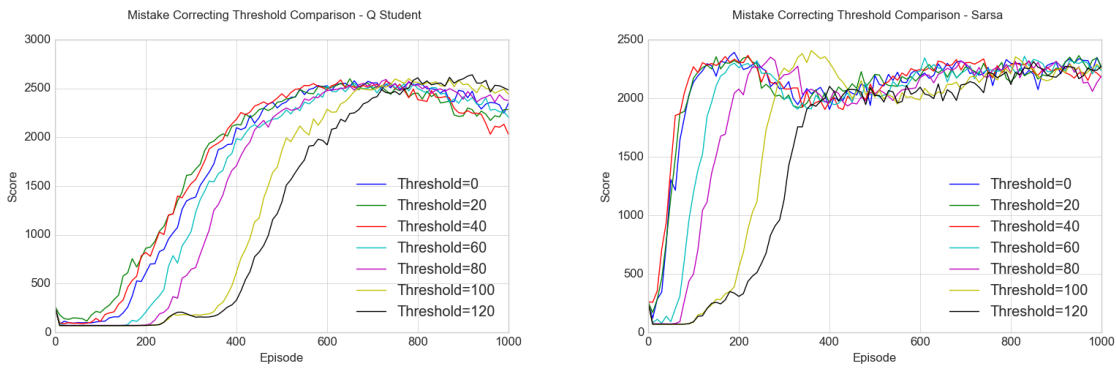


FIGURE 5.2: Threshold comparison for Mistake Correcting, for Q-student and SARSA-student

Similar to Taylor’s results, the teacher-aided student agent improves its performance much more quickly than the unaided student, with the mistake-correcting algorithm appearing to be more effective than advise-important. For the Q-student, I also notice that adding Mistake Correcting teachers appeared to slightly decrease the student’s performance after around 800 episodes, perhaps as a result of over-advising. These results are displayed in Figure 5.3.

From these results, I notice the unaided Q-student agent appears to reach optimal performance by 800 training episodes, and the unaided SARSA-student appears to reach optimality by 600 training episodes, so all the following figures will only show up to those respective numbers of episodes, unless otherwise noted.

5.2 Multi-Teacher Heuristics

Extending the single-teacher model, I obtained multiple teachers by training independent student agents for varying amounts of training. For the student to process multiple potentially different advised actions, I use the majority voting model similar to the one used in the grand-teacher algorithm by Zhan, Anmar, and Taylor [2]. In this model, the teachers share a budget,

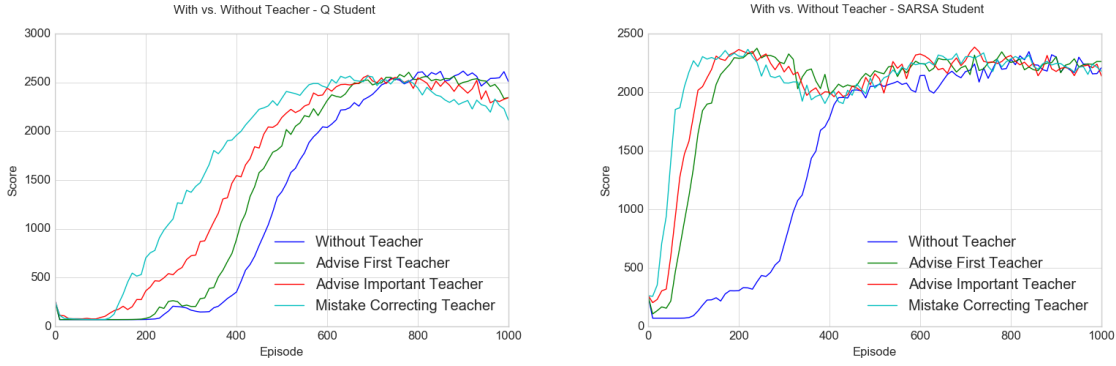


FIGURE 5.3: Comparison of one or no teacher, for Q-student and SARSA-student

and the student chooses the majority action advice given. However, because I focused on teachers with a limited budget, I investigated a couple algorithms for teachers to choose when to give advice, Majority Vote and Independent Teachers.

5.2.1 Choosing When to Advise - Majority Vote

I began with the Majority Voting for choosing when to advise. For each new state the student observes, a strict majority of the teachers must find the state important before they advise.

5.2.1.1 Similar Teachers

I obtained the policies of multiple similar teachers by taking the top five evaluated teachers from the same pool of the independent students trained for the single-teacher model (such that each teacher was trained for 1000 episodes). Teacher 1 is the same teacher as from the single teacher experiments, so it is the best-performing teacher of the five.

First, I compared different importance thresholds for each teacher for Importance Advising and Mistake Correcting, and I found that a threshold of 40 appeared to be most effective not just for Teacher 1 but for all teachers. Because all the teachers were trained for the same length of time, I expected the threshold to be similar for all of them, so I proceeded with a set threshold of 40 for all teachers in both heuristics.

I trained every new student agent with a varying number of teachers (1 to 5), with the teachers being added sequentially (for instance, Teacher 1 and Teacher 2 were used for the run with two teachers, and all but Teacher 5 were used for the run with four teachers), for the heuristics of Early Advising (Figure 5.4), Importance Advising (Figure 5.5), and Mistake Correcting (Figure 5.6). Results for both $Q(\lambda)$ and $SARSA(\lambda)$ students are shown.

I notice that the Q-learning student consistently appeared to benefit the most from more than 2 teachers. For Early Advising, one teacher was sufficient to boost learning earlier, at around

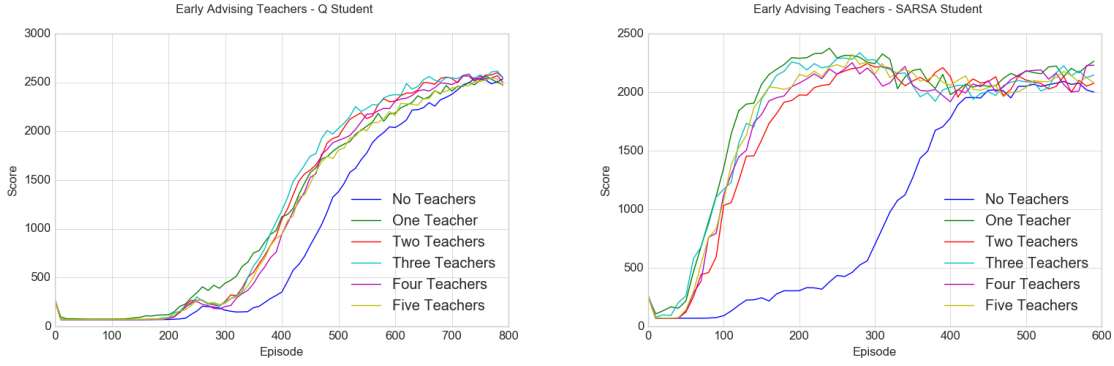


FIGURE 5.4: Comparison of 0-5 Early Advising teachers, for Q-student and SARSA-student

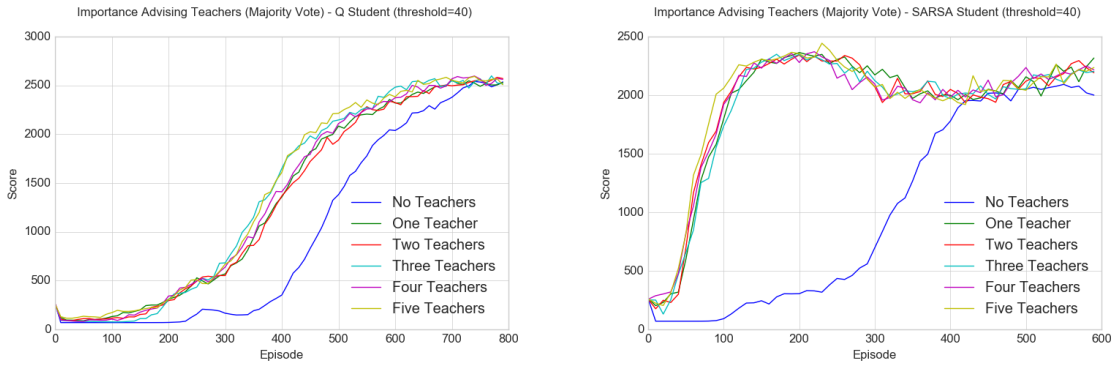


FIGURE 5.5: Comparison of 0-5 Importance Advising teachers, for Q-student and SARSA-student

250 training episodes, but three teachers began to outperform the single teacher at around 400 episodes (see Figure 5.4). For Importance Advising, three and five teachers begin outperforming the other teacher sizes at around 300 episodes (see Figure 5.5), while for Mistake Correcting, one teacher clearly does not perform as well as 2-5 teachers, beginning at around 100 episodes (see Figure 5.6). The SARSA-learning student also appeared to benefit more from greater than 2 teachers, with performance between numbers of teachers being slightly harder to distinguish. An exception is Early Advising, where the SARSA student appeared to learn best with just Teacher 1. Besides this case, it appears multiple similar teachers outperform the single best teacher for all three heuristics.

Interestingly, I also notice that the Q-learning student's learning appears to begin suffering with Mistake Correcting at around 800 training episodes, both for the single teacher and for multiple teachers, and extending the training to 1000 episodes in Figure 5.6 delineated this relationship further. This could be explained as an instance of over-advising; a stricter budget might avoid this negative effect.

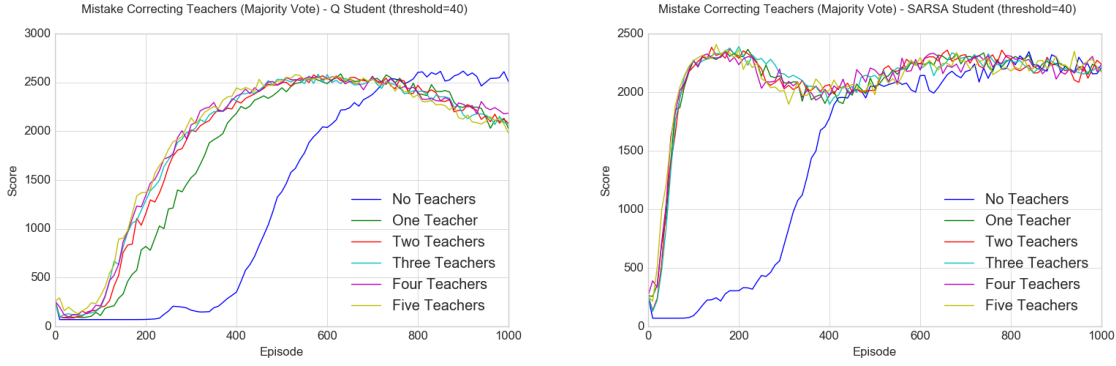


FIGURE 5.6: Comparison of 0-5 Mistake Correcting teachers, for Q-student and SARSA-student

5.2.1.2 Similar Teachers Revisited

Because Teacher 1 was the best-performing teacher, and some students sometimes appeared to perform better with just this best teacher, I wanted to see if the results would change if that teacher was not added first. Thus, I repeated the trials for the multiple similar teachers, but I reversed the order the teachers are added (for instance, Teacher 5 and Teacher 4 were used for the run with two teachers, and all but Teacher 1 were used for the run with four teachers). Results for both $Q(\lambda)$ and $SARSA(\lambda)$ for Early Advising (Figure 5.7), Importance Advising (Figure 5.8), and Mistake Correcting (Figure 5.9) are included.

All students benefit more from multiple teachers in all heuristics, though with no large distinction between the number of teachers in any heuristic. Notably though, contrasting with the previous results, the SARSA-student now appears to benefit more from two teachers instead of one teacher in Early Advising (see Figure 5.7). In addition, this time the Q-student receives an initial boost from three Importance Advising Teachers, but then one and two teachers begin to outperform three teachers at around 300 episodes (see Figure 5.8). In this case, it is possible that Teachers 3, 4, or 5 can provide useful advise in the short run, but Teacher 3 is more helpful in later episodes in conjunction with Teachers 1 and 2. These results highlight the variability of combining teachers even when they are trained for similar amounts of time. The Q-student also still suffers from any number of Mistake Correcting teachers after around 800 training episodes (Figure 5.9 extends to 1000 training episodes to highlight this). In general, however, the results did not drastically change by reversing the order of the similar teachers. Multiple similar teachers still appear to outperform single teachers.

5.2.1.3 Diverse Teachers

I also investigated the case where not every teacher is optimized. I trained additional teachers for 400, 500, 600, and 1000 episodes, comparing how they performed with each other. I chose these values because the lone student has different levels of performance at these points in training;

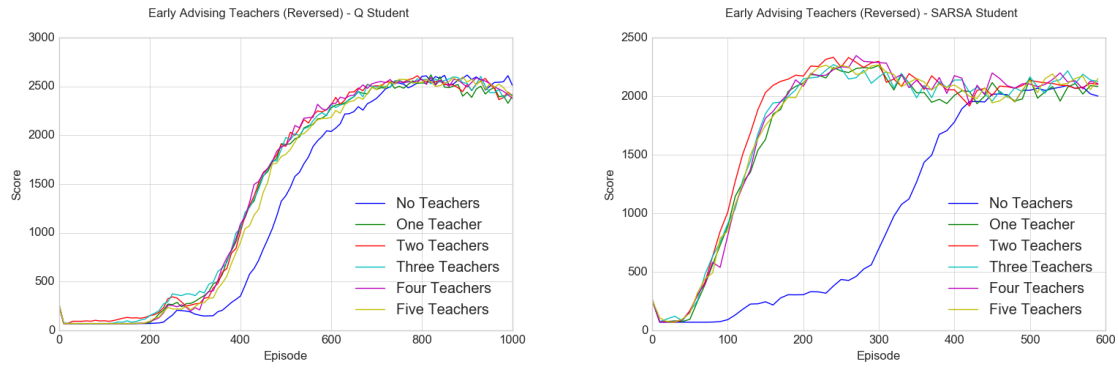


FIGURE 5.7: Comparison of 0-5 Early Advising teachers, for Q-student and SARSA-student

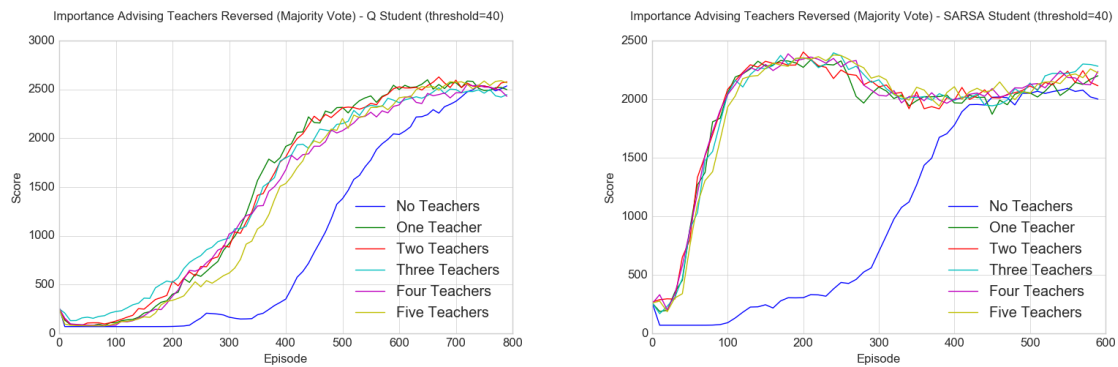


FIGURE 5.8: Comparison of 0-5 Importance Advising teachers, for Q-student and SARSA-student

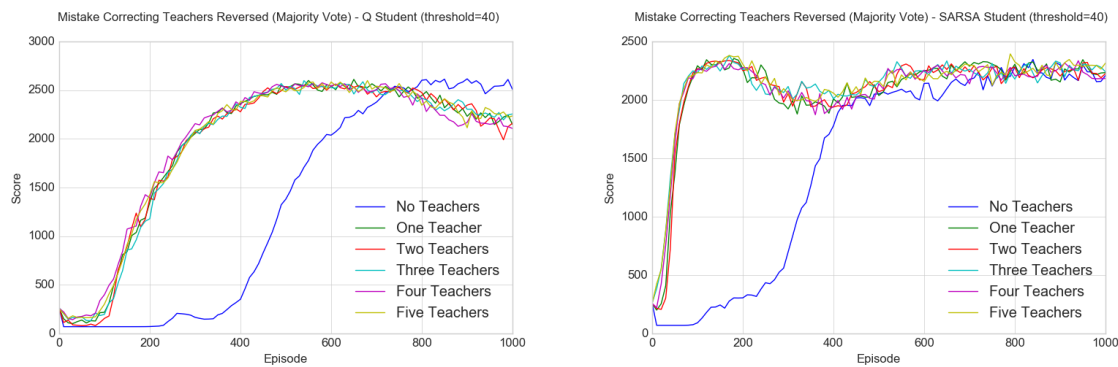


FIGURE 5.9: Comparison of 0-5 Mistake Correcting teachers, for Q-student and SARSA-student

more episodes results in more experience and better performance (see Figure 5.3). Specifically, I grouped the teachers together in different ways to see how better or worse teachers affected the student's learning. The different combinations shown in this section are:

- No teachers
- One teacher - 400 episodes
- One teacher - 1000 episodes
- Two teachers - 400, 500 episodes

- Two teachers - 400, 1000 episodes
- Two teachers - 600, 1000 episodes
- Three teachers - 400, 500, 1000 episodes
- Three teachers - 400, 600, 1000 episodes
- Four teachers - 400, 500, 600, 1000 episodes

First, when looking for the best thresholds, I notice these less-experienced teachers have less range in their importance gaps in general, so I would expect them to struggle more with distinguishing important states from unimportant ones. When I compared different thresholds, the lower thresholds (or a zero threshold) tended to be more effective, so I used a threshold of 20 for all these teachers. This is not the optimal threshold for all the teachers; however, the less-trained teachers are already expected to be less effective teachers.

Once again, results for both $Q(\lambda)$ and $SARSA(\lambda)$ for Early Advising (Figure 5.10), Importance Advising (Figure 5.11), and Mistake Correcting (Figure 5.12) are included. I notice that for the Importance Advising and the Mistake Correcting teachers, the student clearly learns the fastest with the one teacher trained for 1000 episodes, but two teachers trained for 1000 and 600 episodes are the next best for the student and eventually outperform the single teachers. This is especially apparent for the SARSA student, where the two teachers outperform the single teacher at around 250 episodes (see Figures 5.11 and 5.12). However, the areas under the curves of the single teacher are still greater, indicating more utility earlier in the training.

While the two teachers trained for 1000 and 600 episodes could outperform the best teacher, the two teachers trained for 1000 and 400 episodes are not nearly as helpful for the student. In each of the plots, the combination of these teachers results in a learning curve below the other combinations' with the 1000-episode teacher. In fact, it is consistently one of the worst combinations in these results. Similarly, with three Importance Advising or Mistake Correcting teachers, the teachers trained for 400, 600, and 1000 episodes noticeably outperforms teachers trained for 400, 500, and 1000 episodes (see Figures 5.11 and 5.12). This suggests that bad teachers do worse than good teachers, holding other teachers constant. The number of optimal teachers is a bit more ambiguous for the Early Advising teachers - all combinations appear to perform similarly, except the single teacher trained for 400 episodes, the least-experienced teacher, consistently teaches the slowest.

Another interesting result I notice from the Importance Advising and the Mistake Correcting teachers is that certain combinations of teachers are less effective than having no teacher at all. For instance, for the Q-student, all combinations that include the 400-episode or the 500-episode teacher, with the exception of the 400-, 600-, and 1000-episode combination, are outperformed

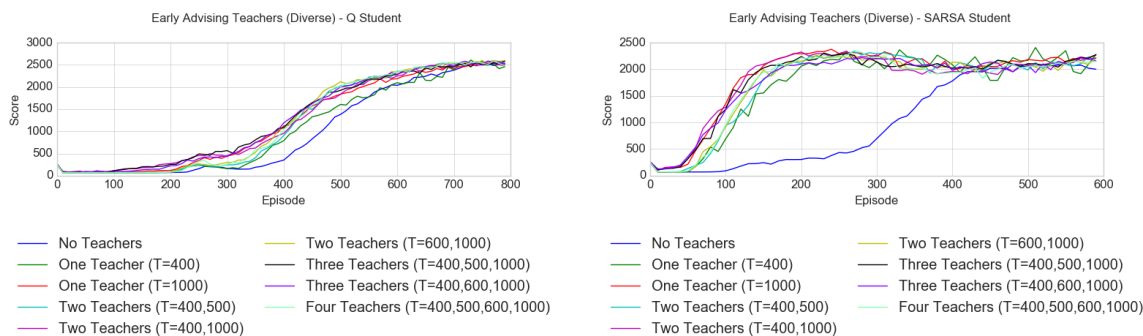


FIGURE 5.10: Comparison of diverse Early Advising teachers, for Q-student and SARSA-student

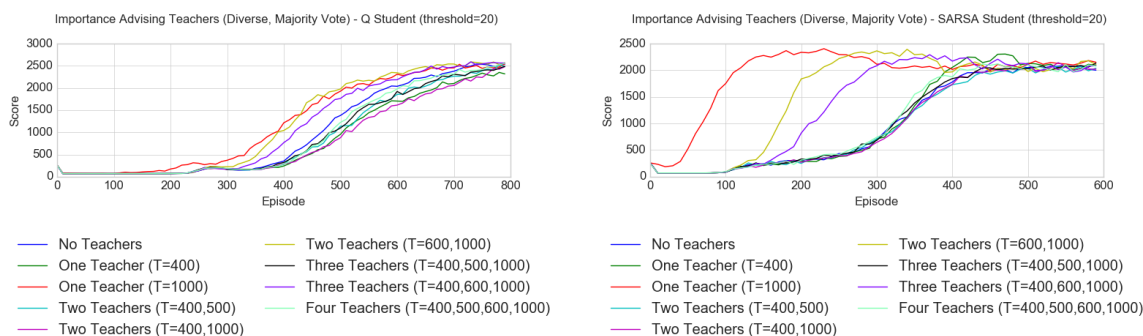


FIGURE 5.11: Comparison of diverse Importance Advising teachers, for Q-student and SARSA-student

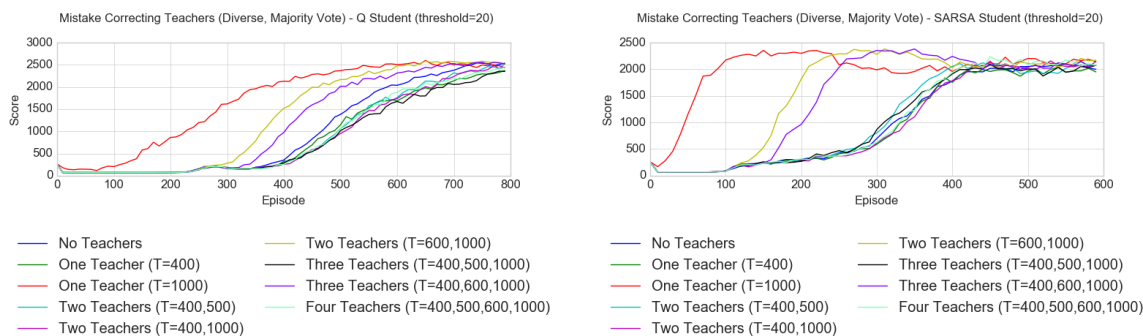


FIGURE 5.12: Comparison of diverse Mistake Correcting teachers, for Q-student and SARSA-student

by the lone student. This appears to show that if at least half of the teachers are bad (in this case the 400- and 500-episode teachers), then the student's learning will suffer.

5.2.1.4 Quantifying Budget Usage

Next, I investigated how teachers were using their budget. More specifically, I wanted to see if the universal budget for all teachers hinders each teacher's individual efficacy. In order to

do this, I looked at when the Importance Advising and Mistake Correcting teachers agreed to advise the student, plotting the cumulative count of this happening for each new student. I also looked at the importance gaps the individual teachers calculated for each state the student visited, and I plotted the cumulative counts of the times the teachers found states important. Note that Early Advising teachers use up their budget in the first b steps of every student's learning, if b is the budget size, so the cumulative plot of the budget usage would be a straight line. Thus budget usage plots are not included for Early Advising teachers.

This approach results in many non-decreasing lines, such as in Figure 5.14. Each line on the plot represents a single episode. As the student traverses different states in different episodes during its learning, the teachers give advice for those states as they are explored, so each episode is expected to result in different budget usages, depending on the path of states the student takes. These plots are titled as "Actual Budget Usage." For comparison, I also tracked when each teacher individually considered a state important for every student path (and thus votes to advise) in the plots titled as "Individual Budget Usage." I did not impose a limit on the individual usage in order to see the trajectory of each teacher uninhibited by the overall budget. While somewhat cluttered, these plots can give a general idea of when and how often teachers use up their budgets, without losing information due to aggregation.

5.2.1.5 Budget Usage - Similar Teachers (Reversed)

First, I took the reversed similar teachers list and plotted the budget usage over the course of each episode when teaching a student for the different numbers of teachers.

Plots for one teacher (5.13), two teachers (5.14), three teachers (5.15), four teachers (5.16), and five teachers (5.17) are shown. I notice that there do not appear to be large differences in the patterns budget usages between the heuristics of Importance Advising, and Mistake Correcting, though the slope of budget usage is much steeper for Importance Advising. For instance, in the single teacher case (Figure 5.13), the teacher advises 1000 times in Importance Advising for between 2000 and 5000 steps, with both the Q-student and the SARSA student. However, in Mistake Correcting, the single teacher doesn't use up its budget until episodes reach between 5000 and 25,000 steps for the Q-student and between 10,000 and 40,000 steps for the SARSA student. This makes sense, as Mistake Correcting is more selective in choosing when to advise; it adds another condition onto Importance Advising.

I also notice that as the number of teachers increases, the individual teachers don't contribute as much; they advise for a fewer number of steps. Take Importance Advising for instance: in the two-teacher case (see Figure 5.14) the actual budget is used up by around 4500 steps and each individual teacher votes to advise up to around 2250 times for all the episodes, but in the five-teacher case (see Figure 5.17) the actual budget is used up by around 3250 steps so that

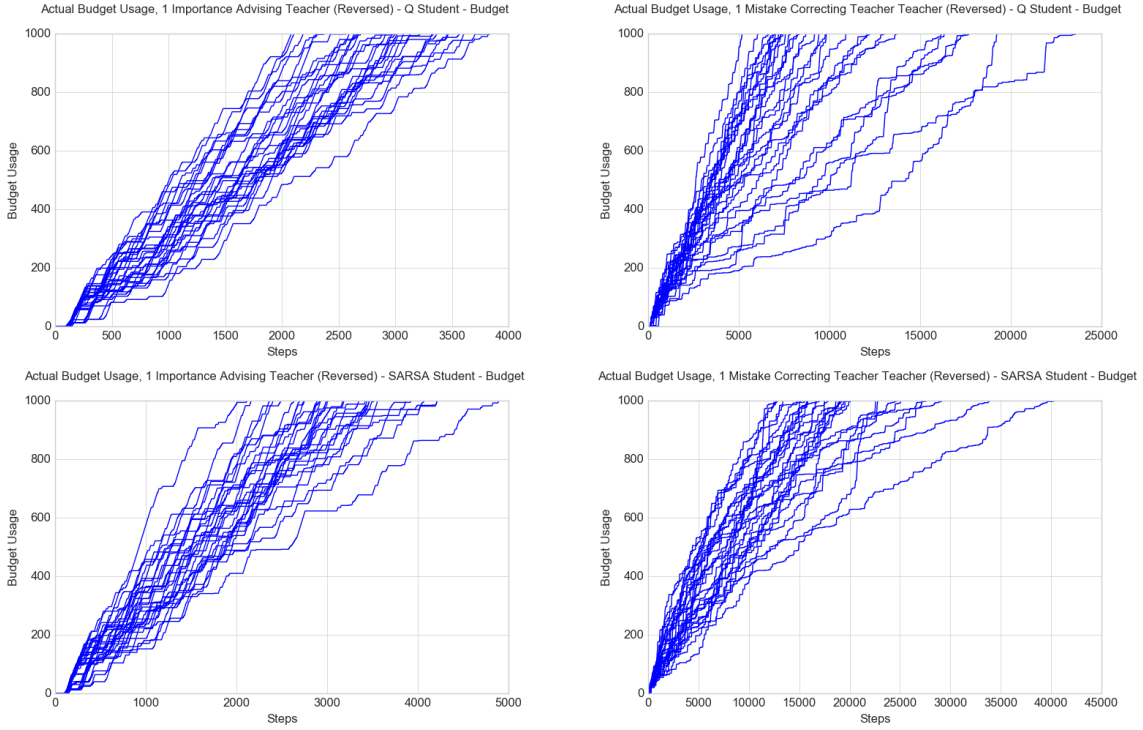


FIGURE 5.13: Comparison of 1 teacher's budget usage (Importance Advising on left and Mistake Correcting on right), for Q-student and SARSA-student

each individual teacher votes to advise up to only around 1300 times for all the episodes. Each teacher is given fewer chances to input individual advice. This makes sense, because the budget is used up when the majority of teachers consider a state important. With more teachers, the majority opinions of the teachers are considered over the individuals.

5.2.1.6 Budget Usage - Diverse Teachers

I also examined the budget usages for diverse teachers. Budget plots of each combination of teachers is displayed (Figures 5.18, 5.19, 5.20, 5.21, 5.22, 5.23, and 5.24). In order to reduce clutter and avoid memory issues when plotting, I limited these plots to a sample of the episodes, so these plots do not necessarily capture information on all the episodes of the student.

The plot for the single teacher trained for 1000 episodes is not displayed, as it is identical to Figure 5.13. Based on these plots, it is clear as to how the bad teachers can hinder learning. The less-trained teachers (such as the teachers trained for 400 or 500 episodes) considered more of the states to be of equal importance, as evident by the smaller range in its important gaps. Thus, with a set threshold of 20, less-trained teachers did not choose to advise as much, as states are less distinct in their policies. This hesitation to advise is apparent in their smaller slopes in the individual budget usage plots. Note that the budget usage is near zero for the teacher trained for 400 episodes for this reason (see Figure 5.18). Similarly, the actual budget usage is

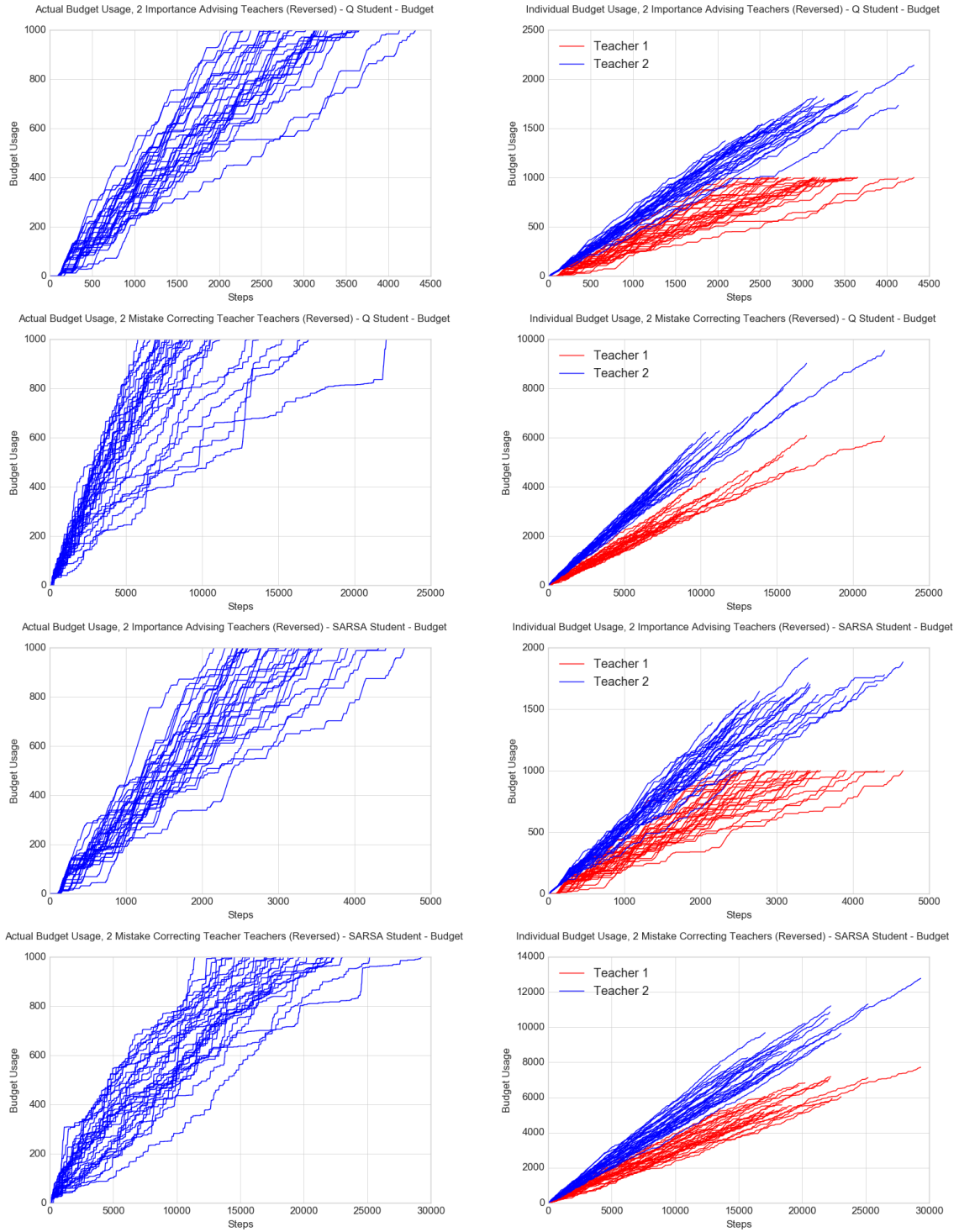


FIGURE 5.14: Comparison of 2 similar teachers' budget usages (Importance Advising and Mistake Correcting), for Q-student and SARSA-student

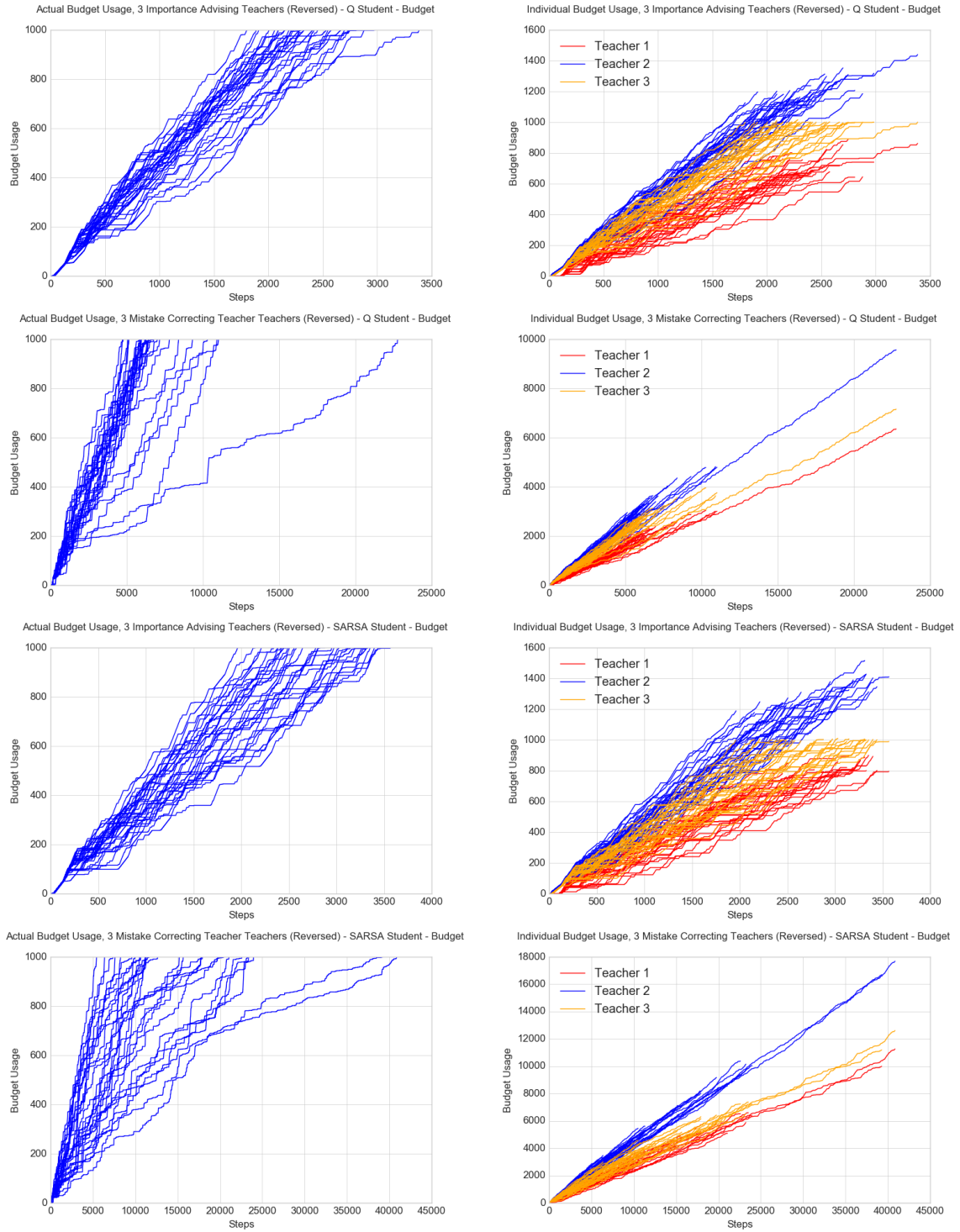


FIGURE 5.15: Comparison of 3 similar teachers' budget usages (Importance Advising and Mistake Correcting), for Q-student and SARSA-student

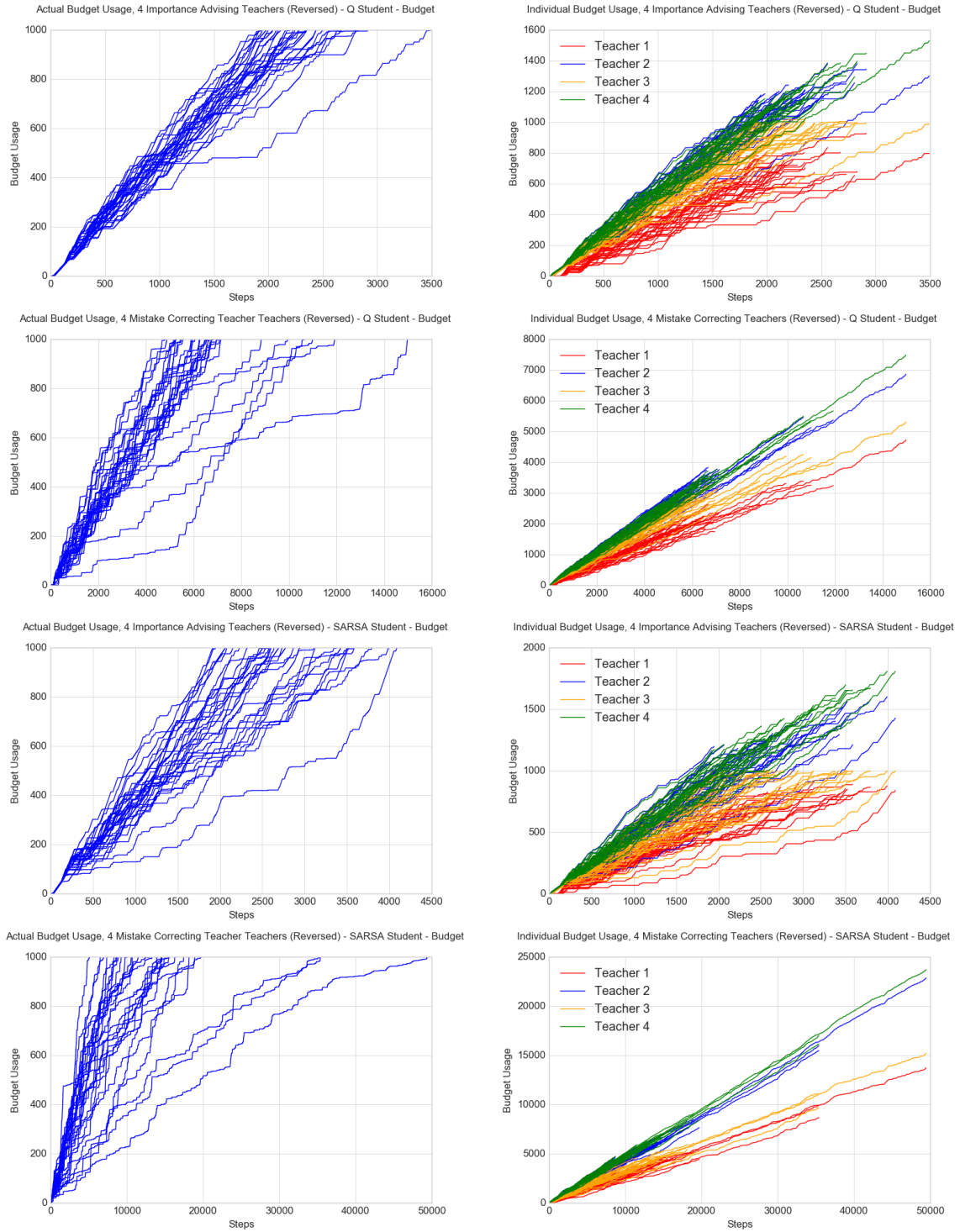


FIGURE 5.16: Comparison of 4 similar teachers' budget usages (Importance Advising and Mistake Correcting), for Q-student and SARSA-student

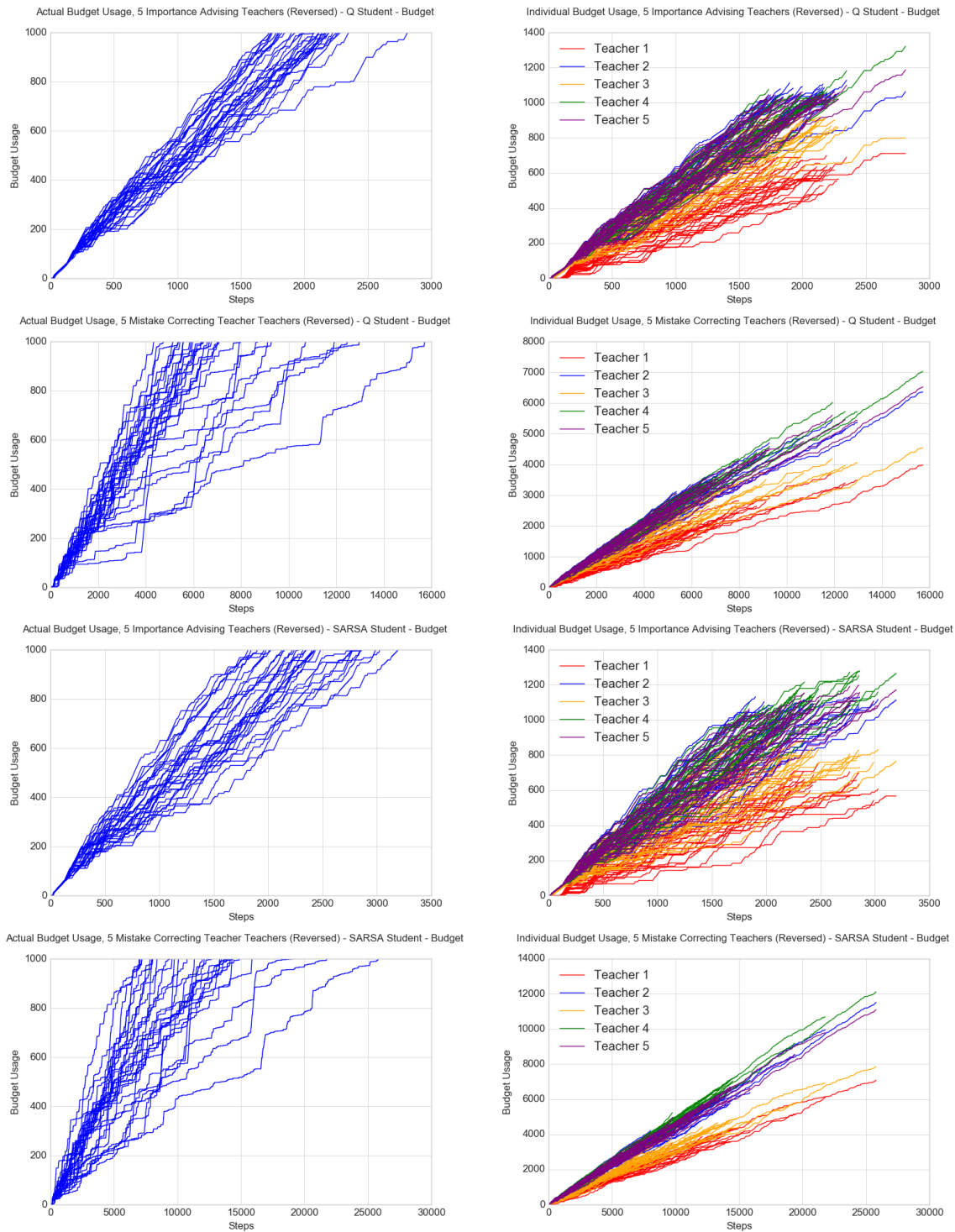


FIGURE 5.17: Comparison of 5 similar teachers' budget usages (Importance Advising and Mistake Correcting), for Q-student and SARSA-student

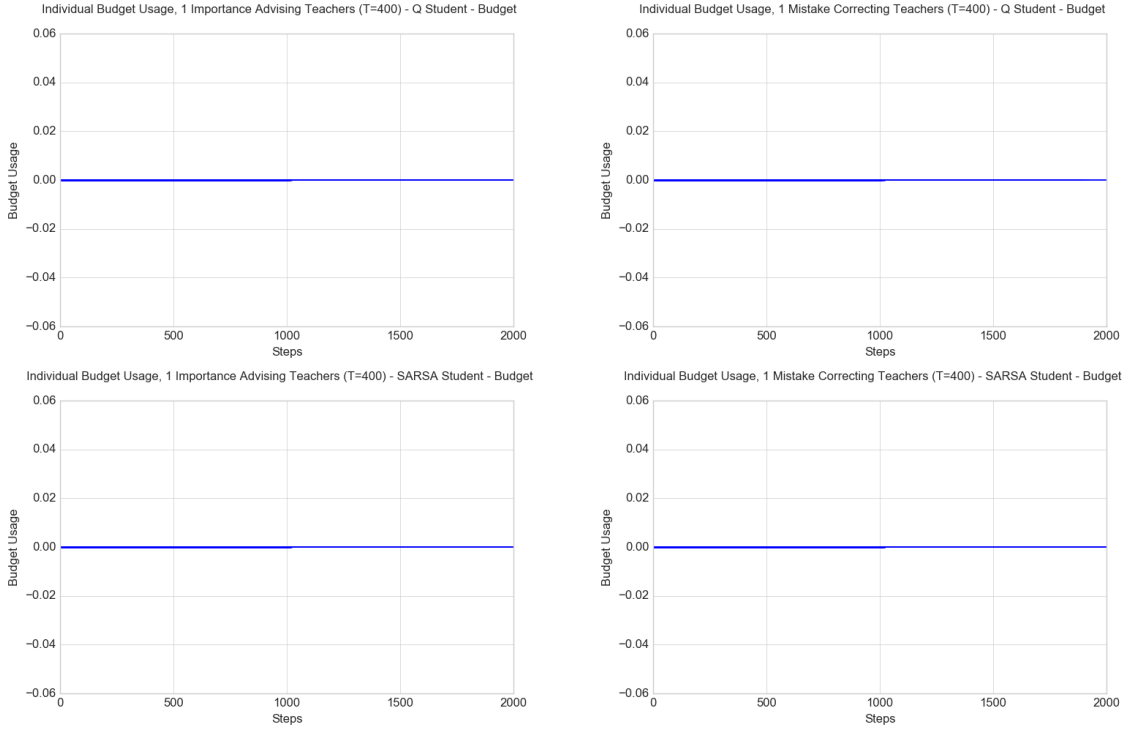


FIGURE 5.18: Comparison of 1 “bad” teacher’s budget usage (Importance Advising on left and Mistake Correcting on right), for Q-student and SARSA-student

near zero for the 400- and 500-episode combination of teachers (see Figure 5.19) as well as the 400- and 1000-episode of teachers (see Figure 5.20), because the less-experienced teacher prevent a majority if it isn’t sure a given state is important. When such “bad” teachers are outvoted by a majority of better teachers, or when the bad teachers do vote to advise (such as Teacher 1 after around 20,000 steps in Importance Advising, Figure 5.21), this effect is mitigated.

These results are similar to the plots seen in the similar teachers case, though the less trained teachers have much lower slopes than more experienced ones, such as with the 600- and 1000-episode teachers (see Figure 5.21), due to the aforementioned gap similarity.

5.2.2 Choosing When to Advise - Independent Teachers

In the hopes of not under-utilizing teachers, I set up a slightly different framework that more heavily leverages each teacher individually.

I repeat the process performed for the Grand-Teacher but this time implementing the independent teacher voting system rather than the majority voting. Because each teacher has its own individual budget ($b_1 = \dots = b_t = 1000$), plots of the budget usage should resemble the individual budget usages from the majority-voting model, such as in Figures 5.14, 5.15, 5.16, and 5.17.

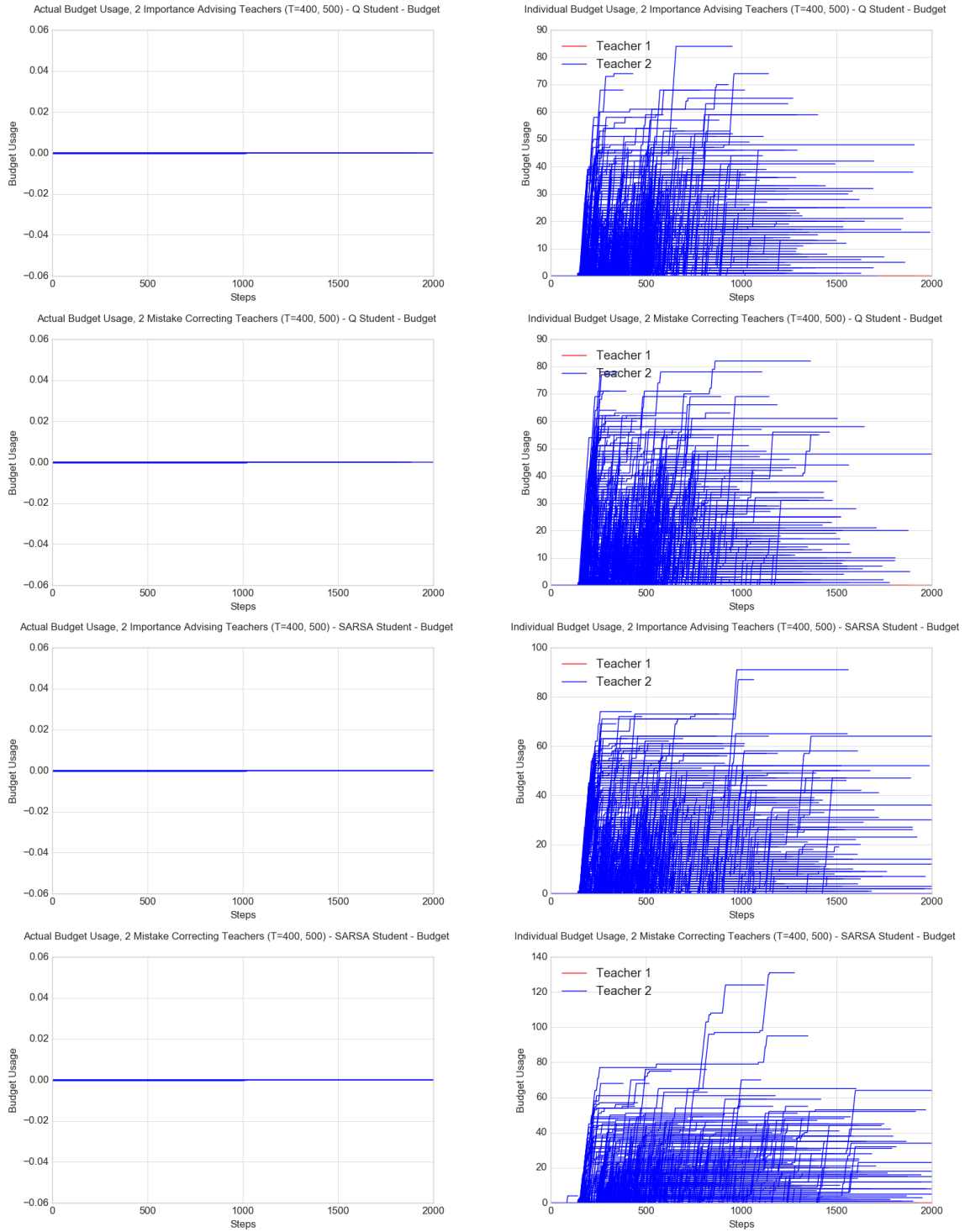


FIGURE 5.19: Comparison of 2 teacher's budget usage (Importance Advising on left and Mistake Correcting on right), for Q-student and SARSA-student

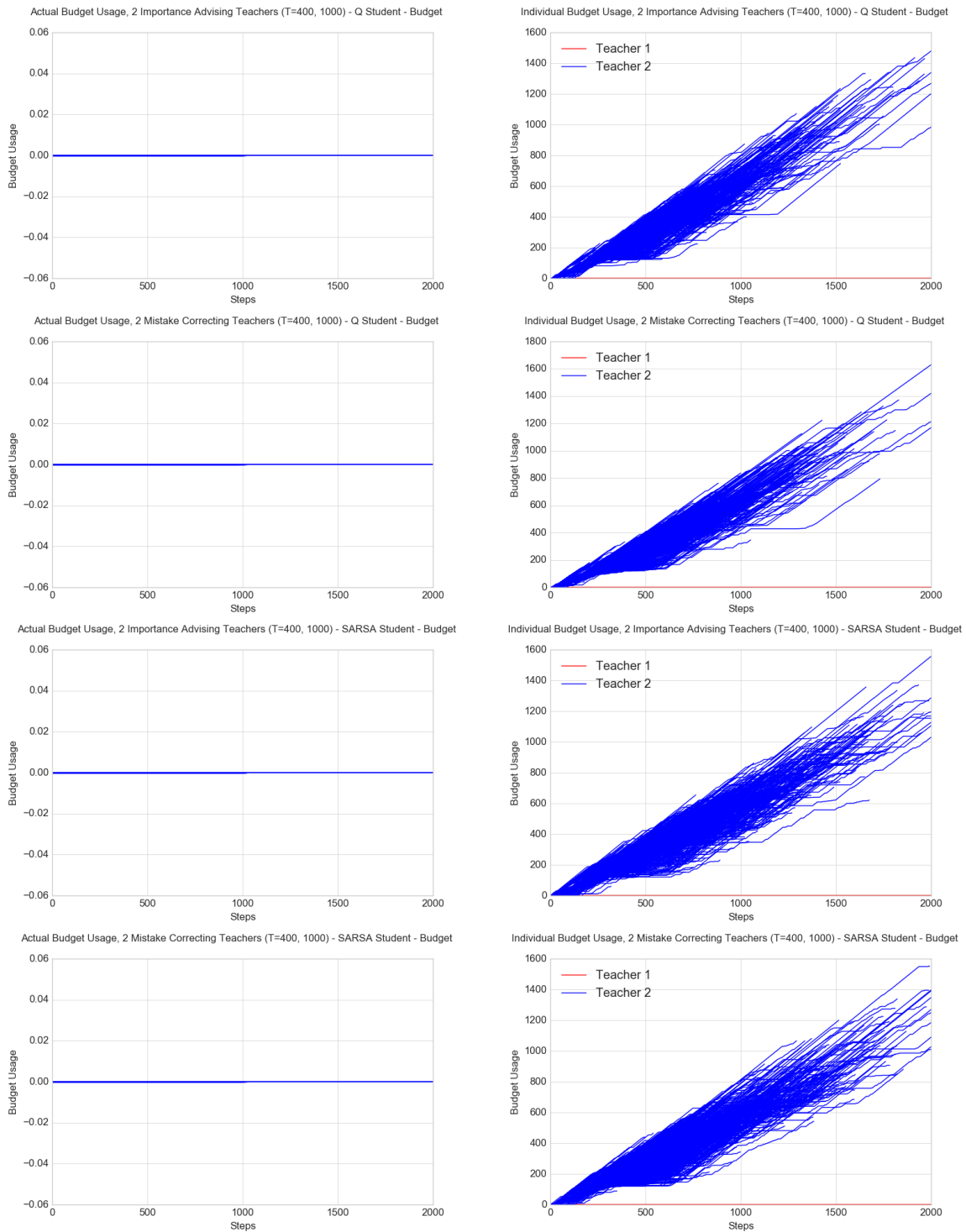


FIGURE 5.20: Comparison of 2 teachers' budget usage (Importance Advising on left and Mistake Correcting on right), for Q-student and SARSA-student

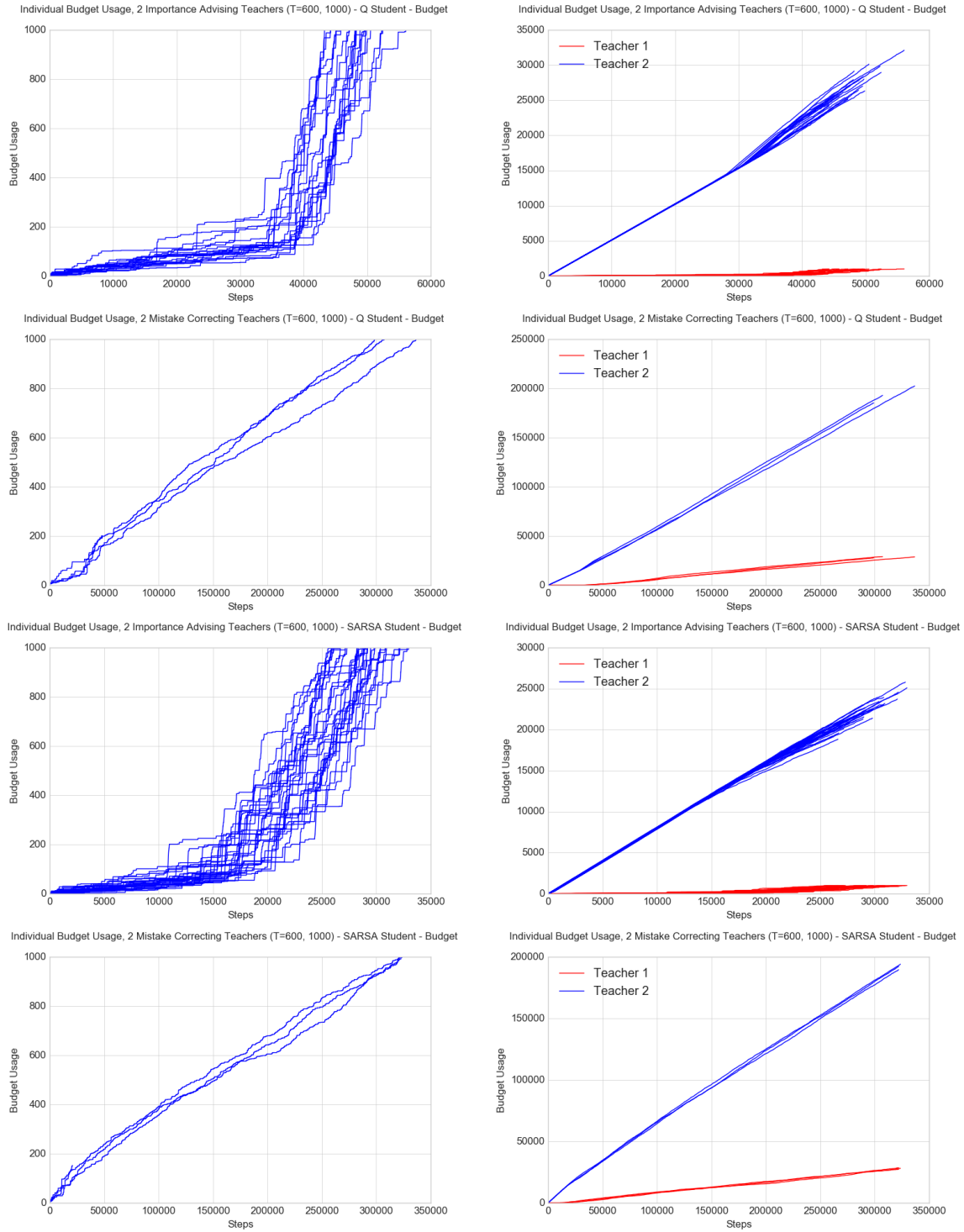


FIGURE 5.21: Comparison of 2 teachers' budget usage (Importance Advising on left and Mistake Correcting on right), for Q-student and SARSA-student

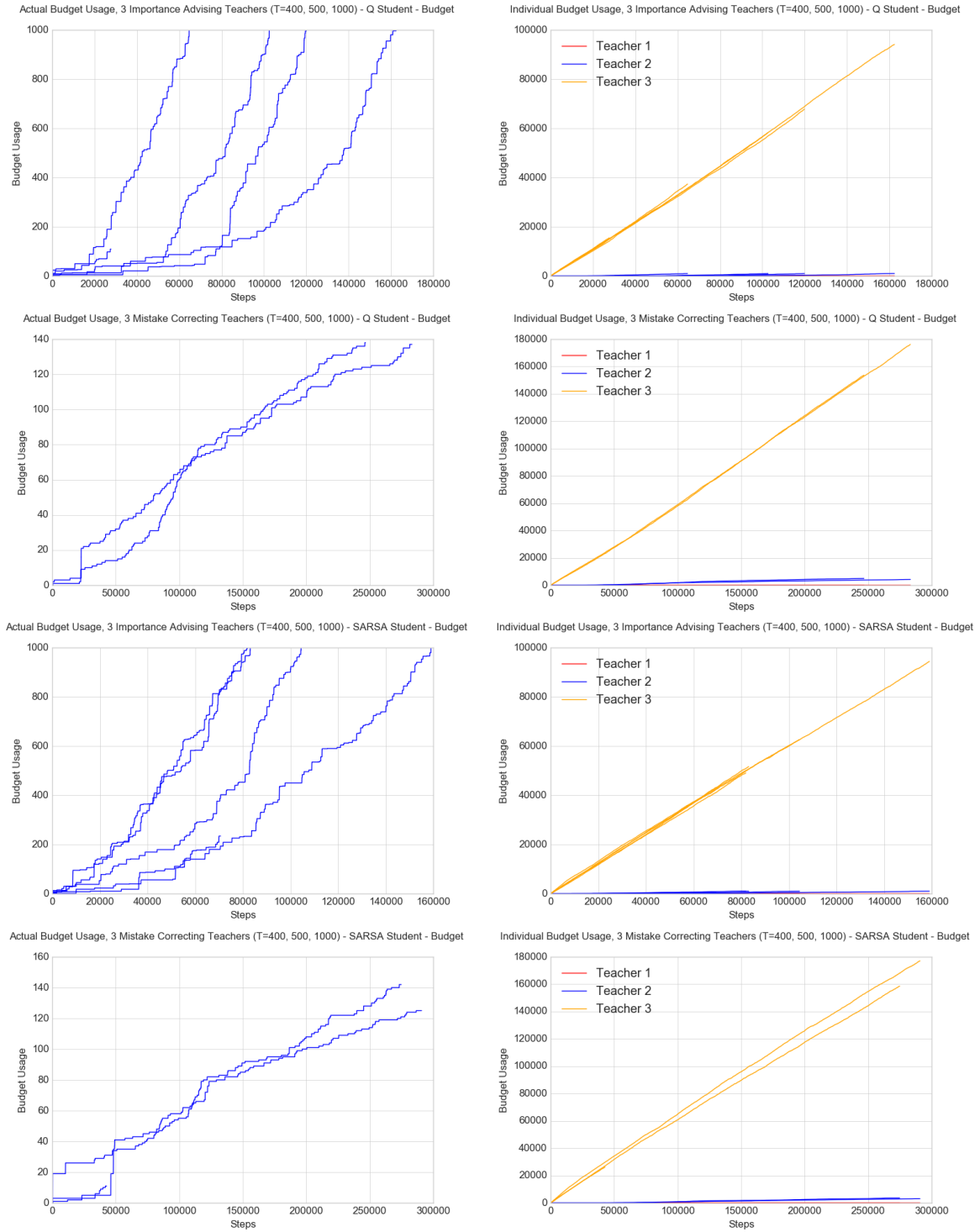


FIGURE 5.22: Comparison of 3 teachers' budget usage (Importance Advising on left and Mistake Correcting on right), for Q-student and SARSA-student

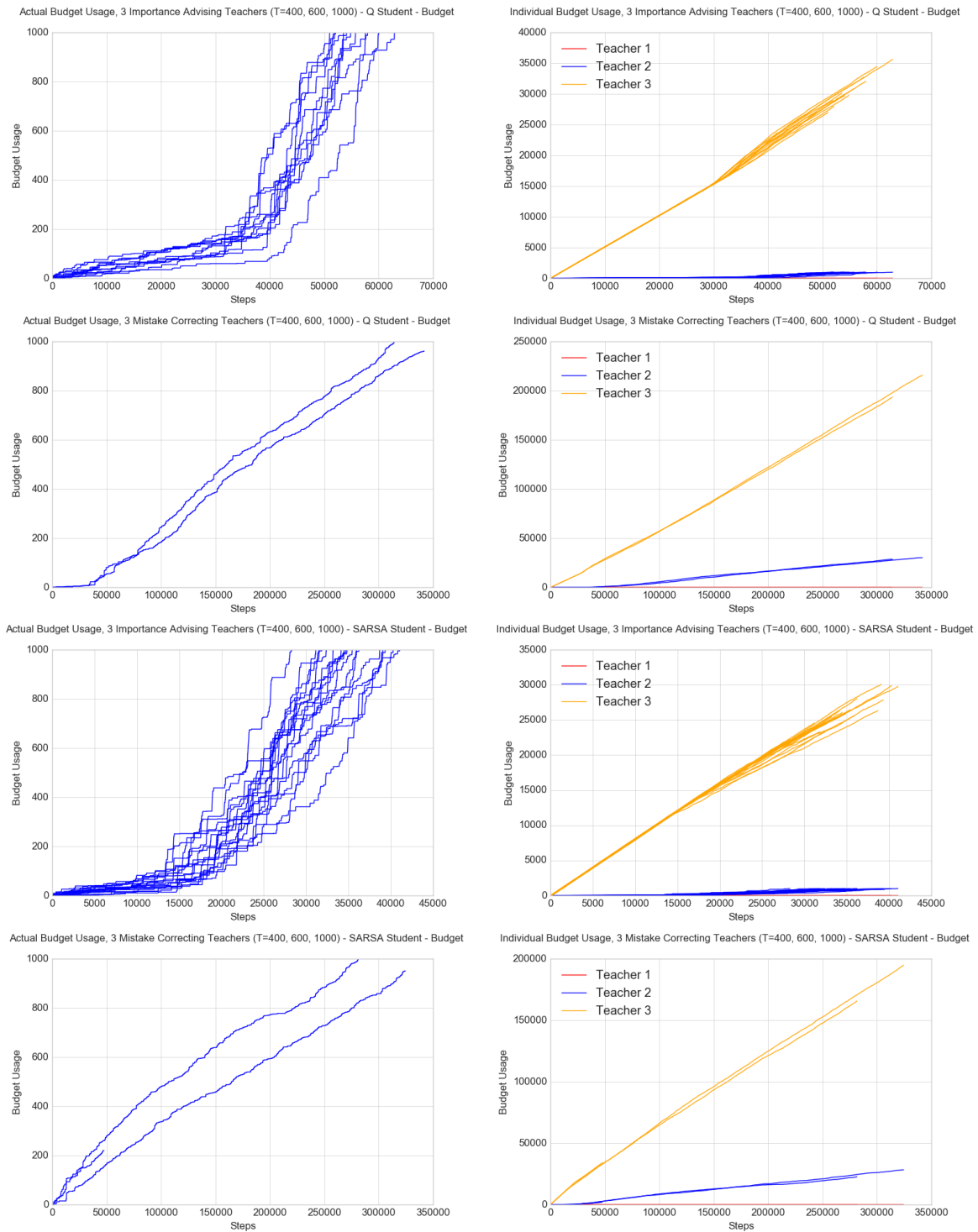


FIGURE 5.23: Comparison of 3 teachers' budget usage (Importance Advising on left and Mistake Correcting on right), for Q-student and SARSA-student

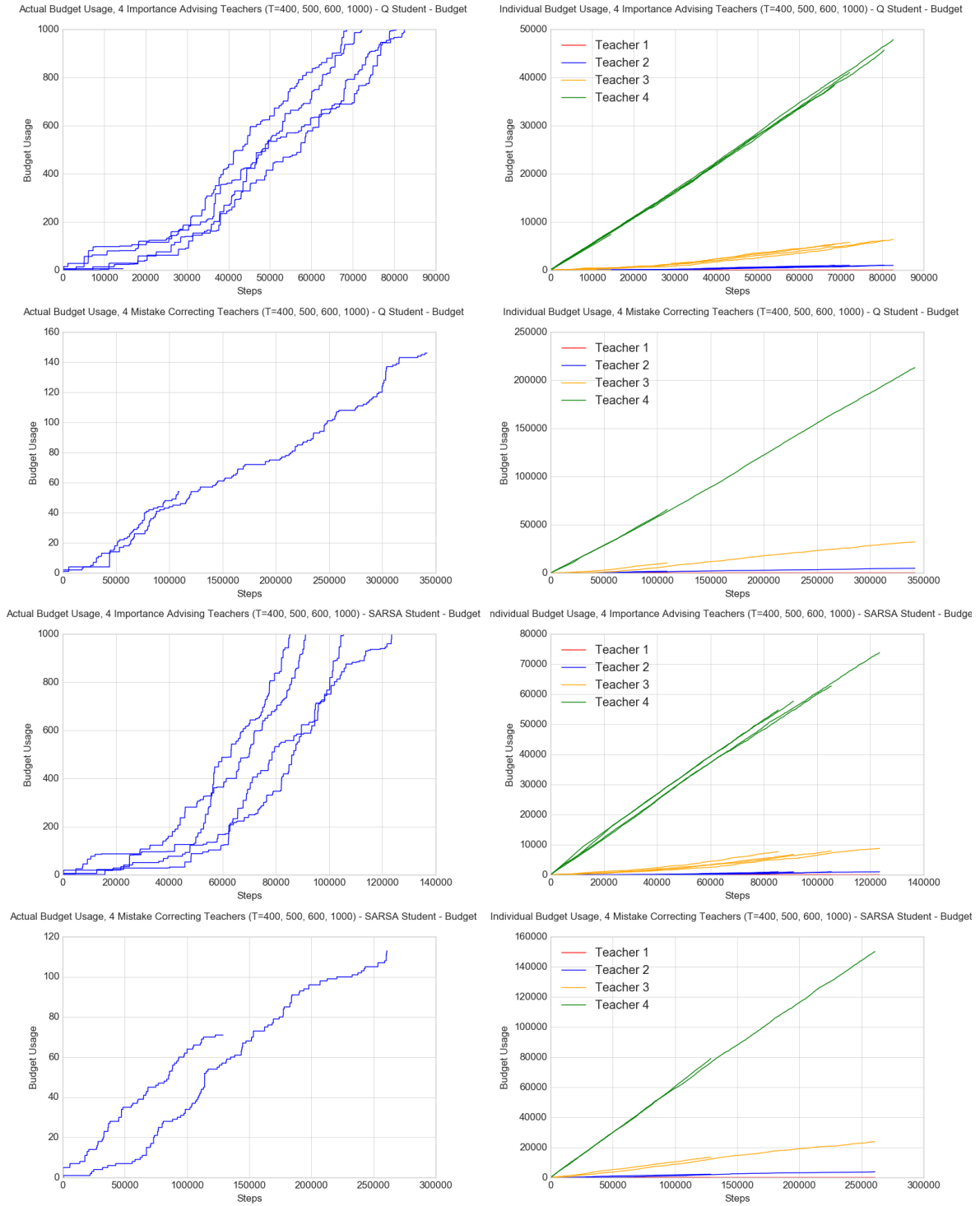


FIGURE 5.24: Comparison of 4 teachers' budget usage (Importance Advising on left and Mistake Correcting on right), for Q-student and SARSA-student

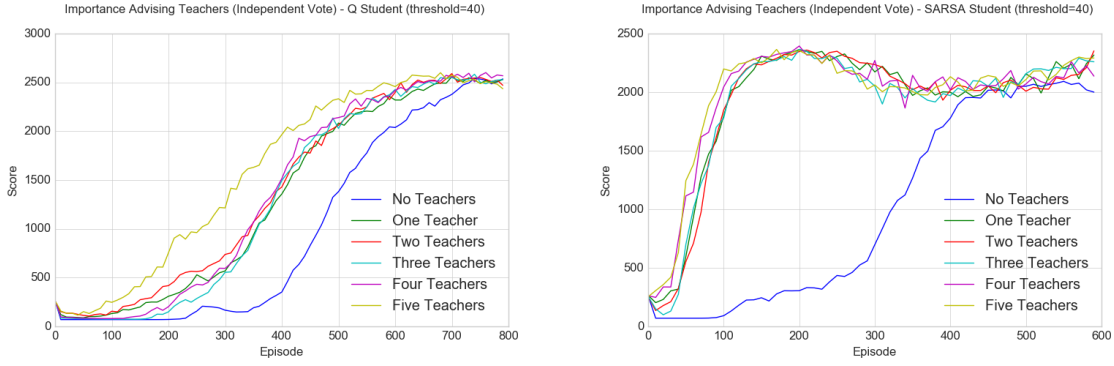


FIGURE 5.25: Comparison of 0-5 independent Importance Advising teachers, for Q-student and SARSA-student

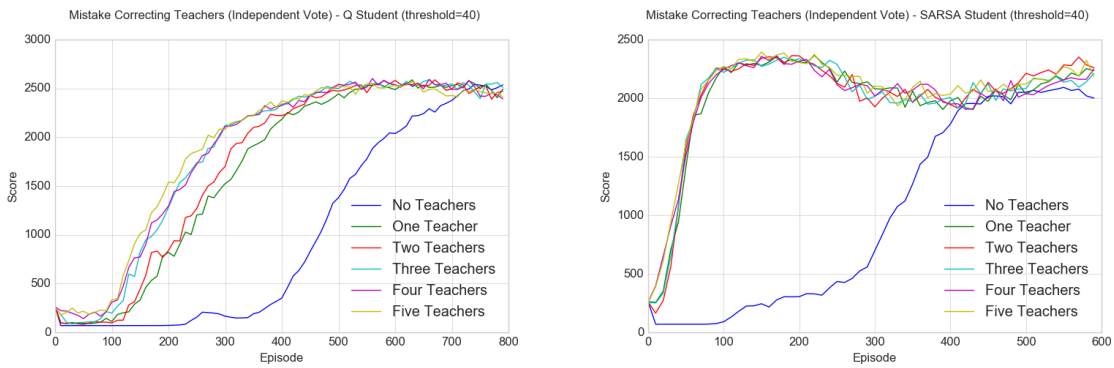


FIGURE 5.26: Comparison of 0-5 independent Mistake Correcting teachers, for Q-student and SARSA-student

5.2.2.1 Similar Teachers

Starting with the same five similar teachers (all trained for 1000 episodes), I repeated the experiment but with the independent-voting framework. Note that Early Advising yields the same results in both the Independent Teachers and the Grand-Teacher models, so I only include the results for Importance Advising and Mistake Correcting. Also note that the algorithms are also identical in the single-teacher case.

Importance Advising (Figure 5.25) and Mistake Correcting (Figure 5.26) plots for 2-5 independent teachers are included, with 0 and 1 teachers for comparison. In all cases, more teachers results in better student performance. This is especially true for Q-students in both Importance Advising and Mistake Correcting (Figures 5.25, 5.26), who learn most quickly from 5 teachers. Results for the SARSA student are very similar for both algorithms.

5.2.2.2 Similar Teachers Revisited

Similarly, I repeat the process for the same teachers in reverse order for comparison.

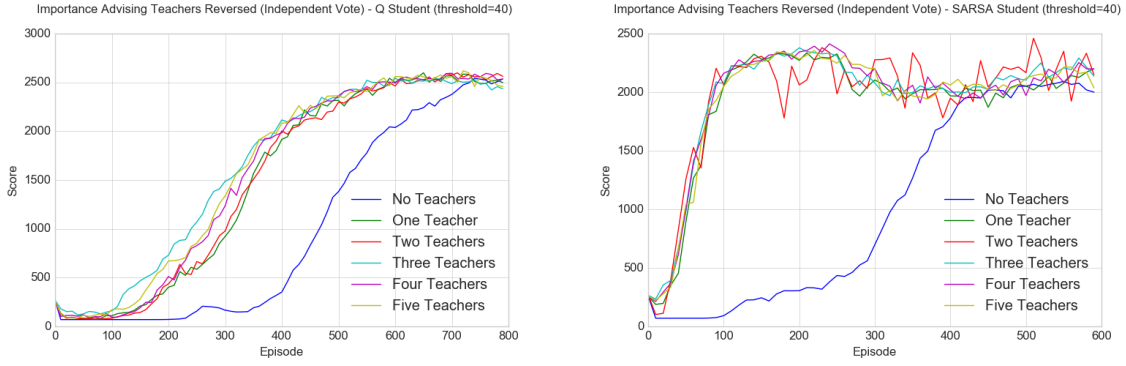


FIGURE 5.27: Comparison of 0-5 independent Importance Advising teachers, for Q-student and SARSA-student

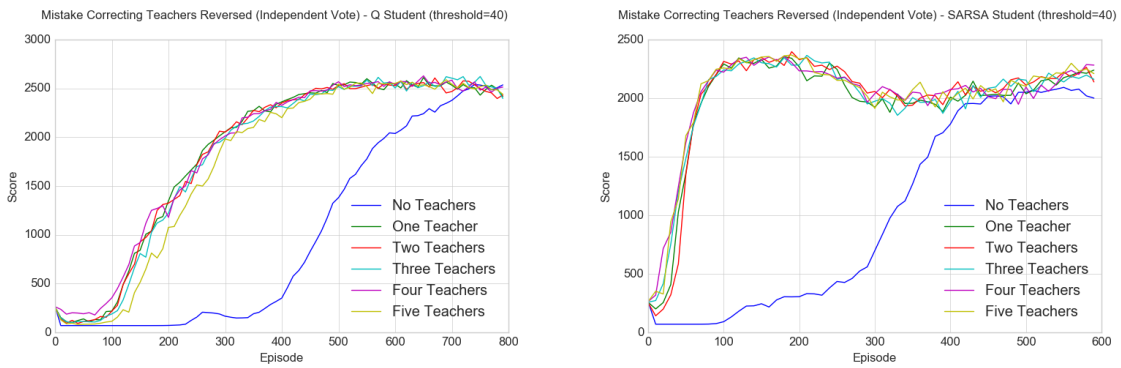


FIGURE 5.28: Comparison of 0-5 independent Mistake Correcting teachers, for Q-student and SARSA-student

Figure 5.27 displays the results for reversed Importance Advising teachers, and Figure 5.28 displays the results for reversed Mistake Correcting teachers. Interestingly, for the Q-students, five teachers appears to be less helpful than the non-reversed teachers; as they are the same set of teachers, it could just happen by chance that these students performed less well than previously, or that the previous students performed better than these.

In general, results are once again very comparable to the same teachers, non-reversed. However, more of the multiple teachers perform better than the single teacher for the Q-student. For instance, in Importance Advising, three to five teachers perform noticeably better than the single teacher (see Figure 5.27). Results for the SARSA student are more similar to the non-reversed results.

5.2.2.3 Diverse Teachers

Repeating for diverse teachers, results for independent Importance Advising (Figure 5.29) and Mistake Correcting (Figure 5.30) teachers are shown. I notice these results are almost identical to the majority-voting diverse teachers, but the four teachers perform slightly better in Importance Advising (see Figure 5.29), and for Mistake Correcting, the two teachers trained for 400 and

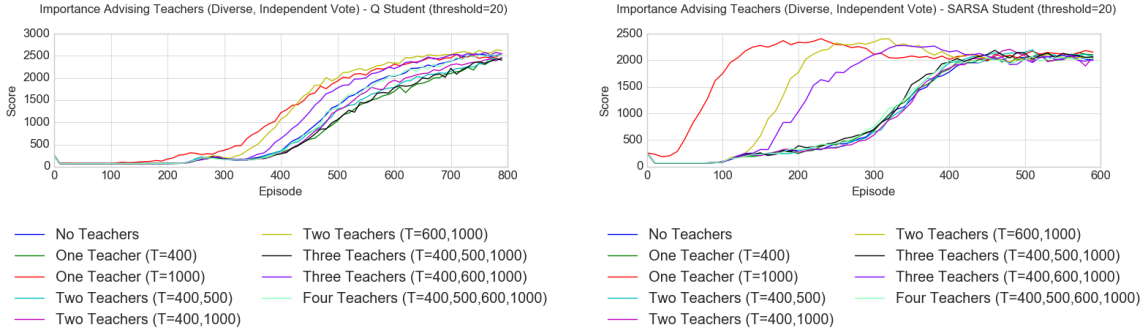


FIGURE 5.29: Comparison of diverse independent Importance Advising teachers, for Q-student and SARSA-student

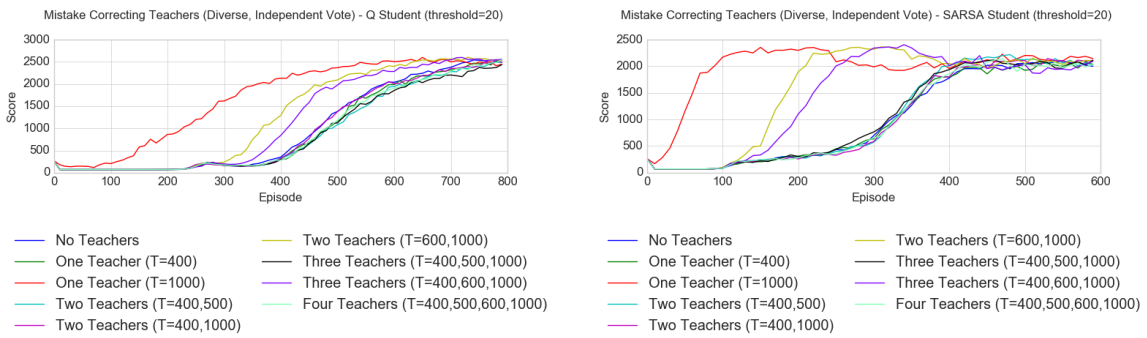


FIGURE 5.30: Comparison of diverse independent Mistake Correcting teachers, for Q-student and SARSA-student

1000 episodes perform slightly better for the Q-student yet slightly worse for the SARSA student (see Figure 5.30). Otherwise, the results are very comparable. This makes sense, because the diverse teachers are expected to have more of an individual effect on learning, even in the majority-voting algorithm. Though bad teachers hinder learning in the majority-voting case, they also slow learning in this independent-voting case as well, as each teacher is guaranteed a voice.

5.2.2.4 Budget Usage - Similar Teachers (Reversed)

Similar to the majority-voting teachers, I display the budget usages for 2-5 reversed teachers, this time for independent teachers (Figures 5.31, 5.32, 5.33, 5.34). As there is no overall budget for a grand-teacher, only the individual budgets are shown. These are immediately comparable to the individual budgets displayed previously. As expected, the budget usage resembles the individual teachers' voting rate in Majority-Voting, though with a cutoff at the budget limit of 1000. As the number of teachers increase, the rate of budget usage over all the teachers does not change much. This is because every teacher independently advises and uses up its budget, so the student will continue receiving advice until the last teacher uses up its budget.

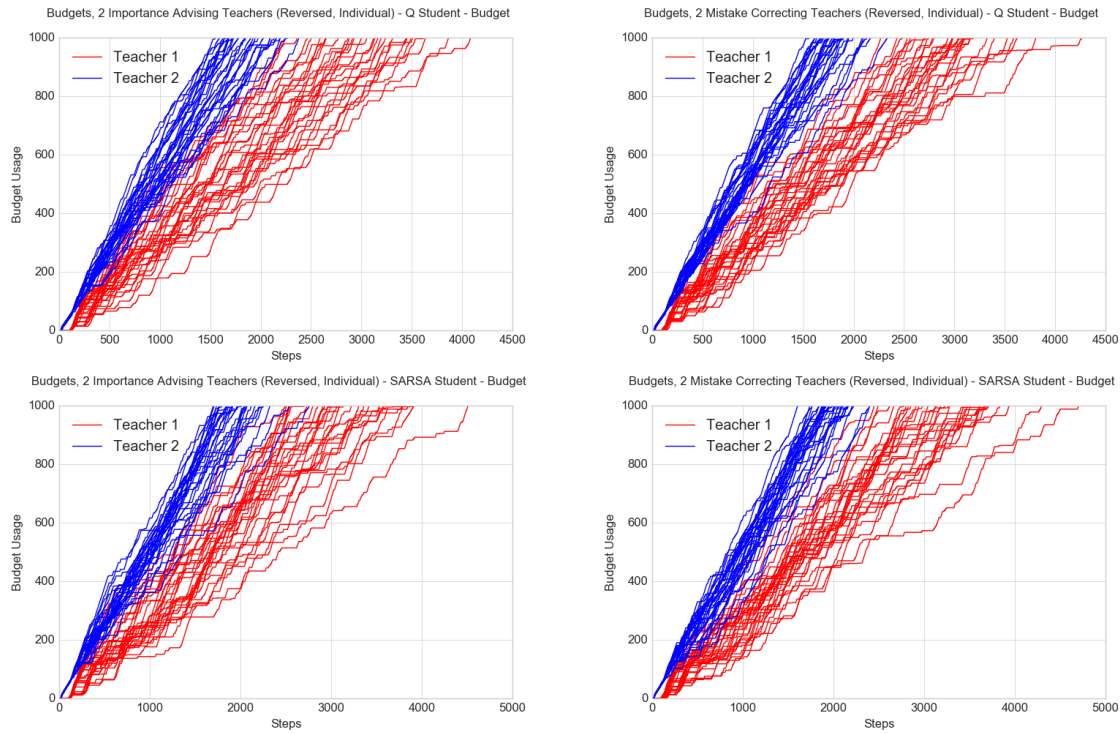


FIGURE 5.31: Comparison of 2 independent teachers' budget usages (Importance Advising and Mistake Correcting), for Q-student and SARSA-student

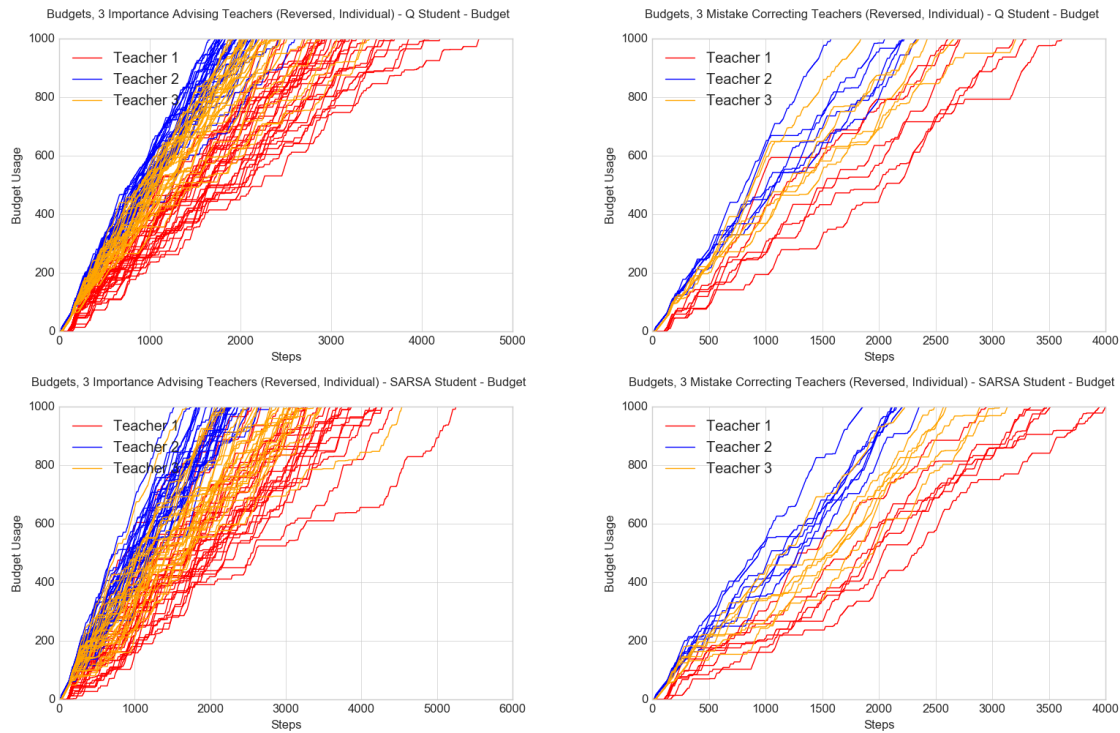


FIGURE 5.32: Comparison of 3 independent teachers' budget usages (Importance Advising and Mistake Correcting), for Q-student and SARSA-student

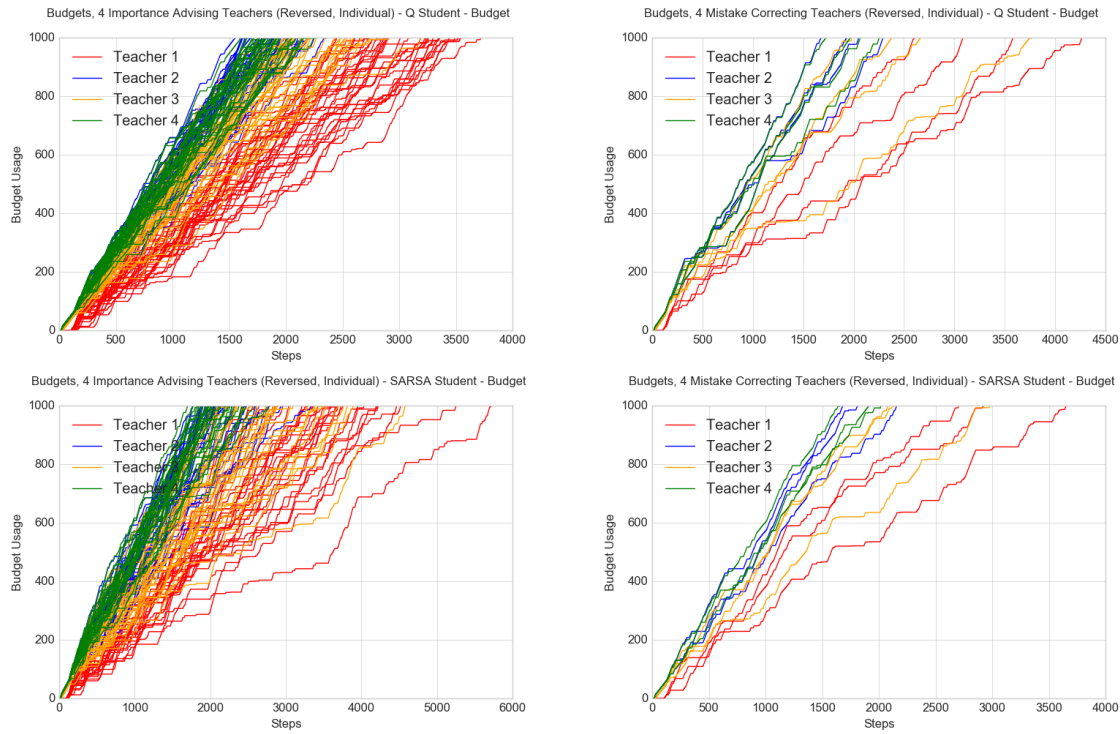


FIGURE 5.33: Comparison of 4 independent teachers' budget usages (Importance Advising and Mistake Correcting), for Q-student and SARSA-student

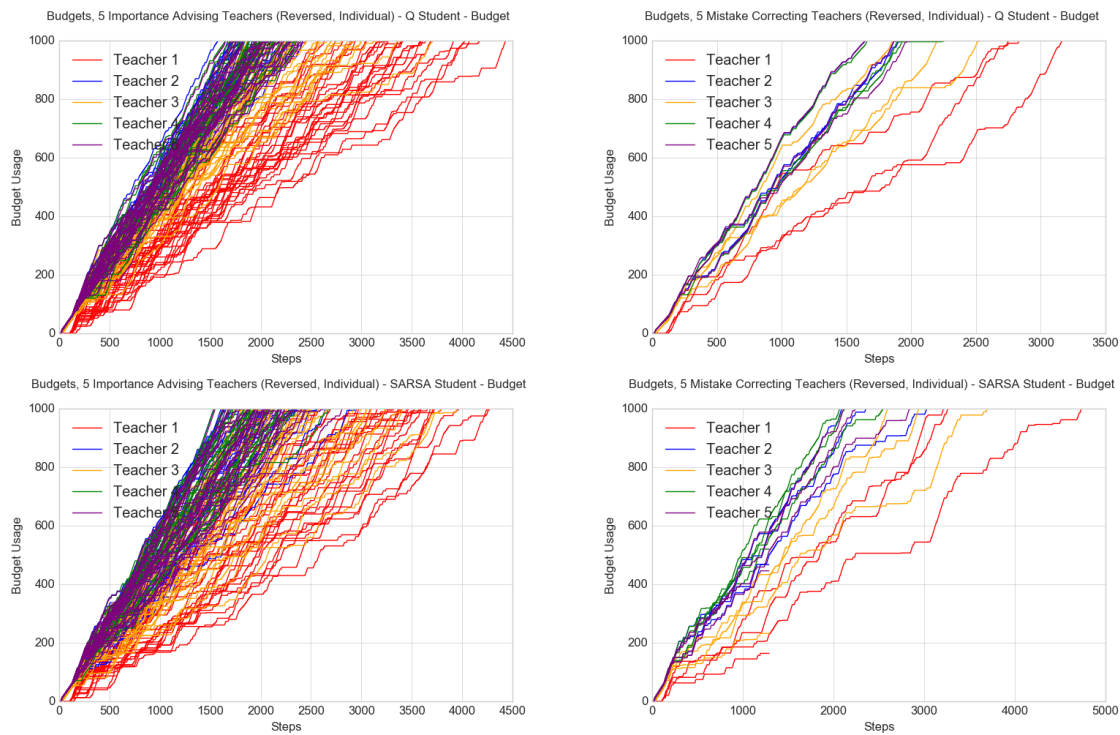


FIGURE 5.34: Comparison of 5 independent teachers' budget usages (Importance Advising and Mistake Correcting), for Q-student and SARSA-student

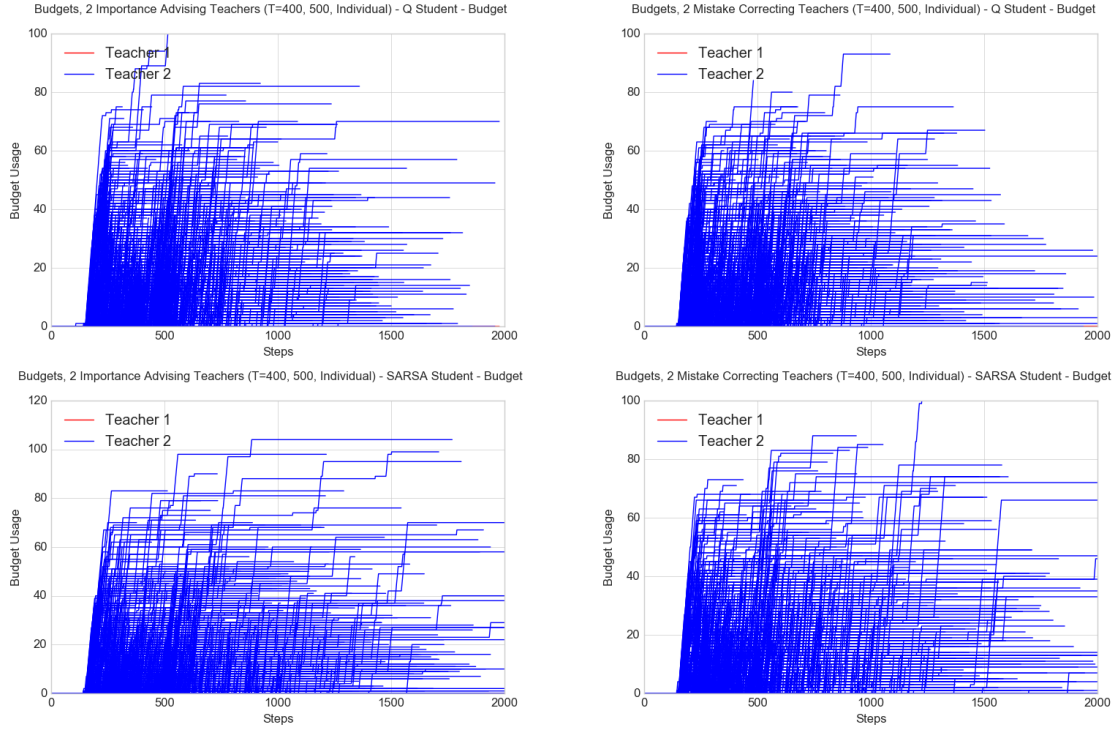


FIGURE 5.35: Comparison of 2 independent teachers' ($T=400, 500$) budget usage (Importance Advising on left and Mistake Correcting on right), for Q-student and SARSA-student

5.2.2.5 Budget Usage - Diverse Teachers

Displayed are the the budget usages for independent diverse teachers (Figures 5.35, 5.36, 5.37, 5.38, 5.39, 5.40). Again, I limited these plots to a sample of the episodes, so these plots do not necessarily capture information on all the episodes of the student. I also did not include the budget usage for one teacher, as it would be identical to Figure 5.18 for the 400-episode teacher and to Figure 5.13 for the 1000-episode teacher.

Similar to the majority-voting case, the less-trained teachers don't advise as often as the teachers who have trained closer to optimality (who have more confidently polarized importance calculations for states). As expected, these plots mirror the patterns seen in the majority-voting individual budgets case. This algorithm does not affect individual importance calculations, but it sets an upper limit on the individual budgets. Compared to the majority-voting algorithm, this method can allow for more advice over a longer period of time.

5.2.3 Comparing Majority Vote with Independent Teachers

Comparing the same teachers for majority voting and independent advising, I check whether the Independent Teachers' learning curves are significantly larger than the Majority Voting teachers' (with 95% confidence). P-values calculated from a one-sided t -test are shown for

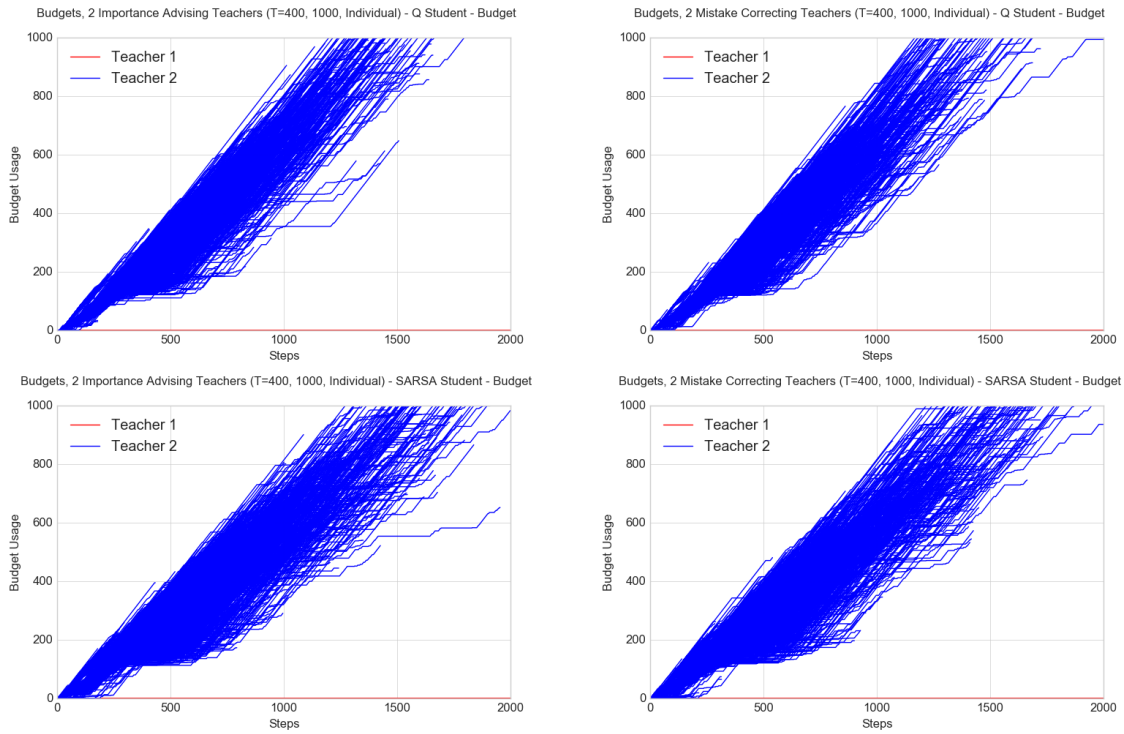


FIGURE 5.36: Comparison of 2 independent teachers' (T=400, 1000) budget usage (Importance Advising on left and Mistake Correcting on right), for Q-student and SARSA-student

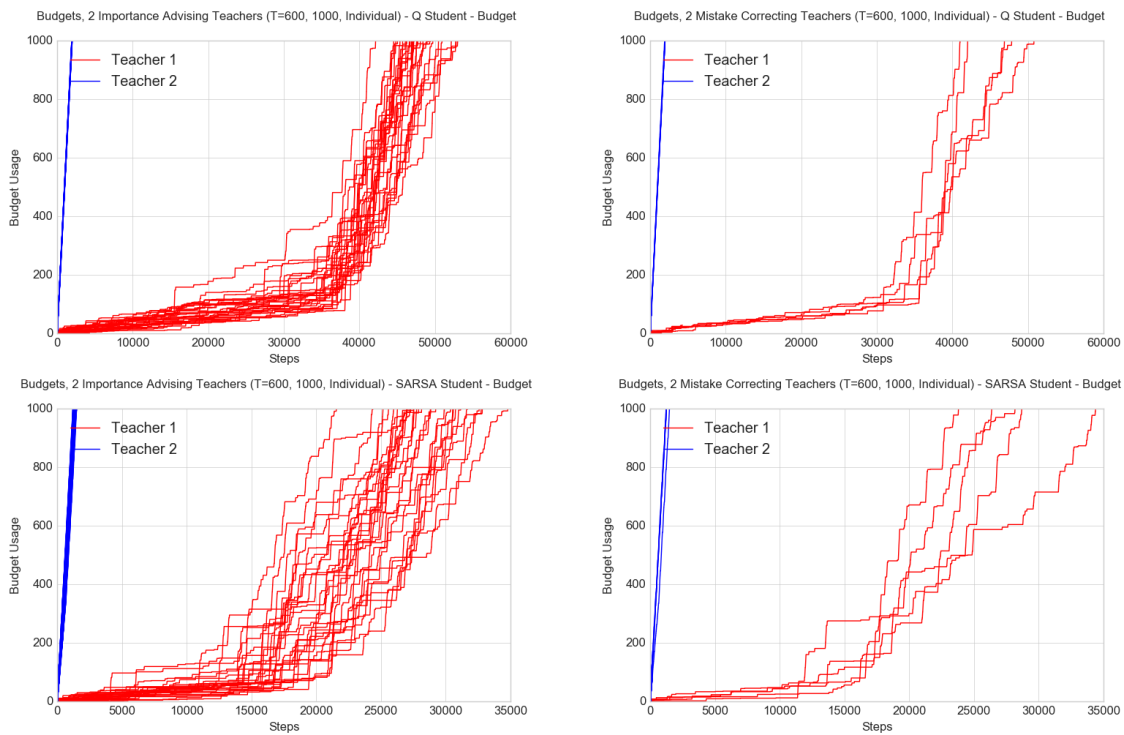


FIGURE 5.37: Comparison of 2 independent teachers' (T=600, 1000) budget usage (Importance Advising on left and Mistake Correcting on right), for Q-student and SARSA-student

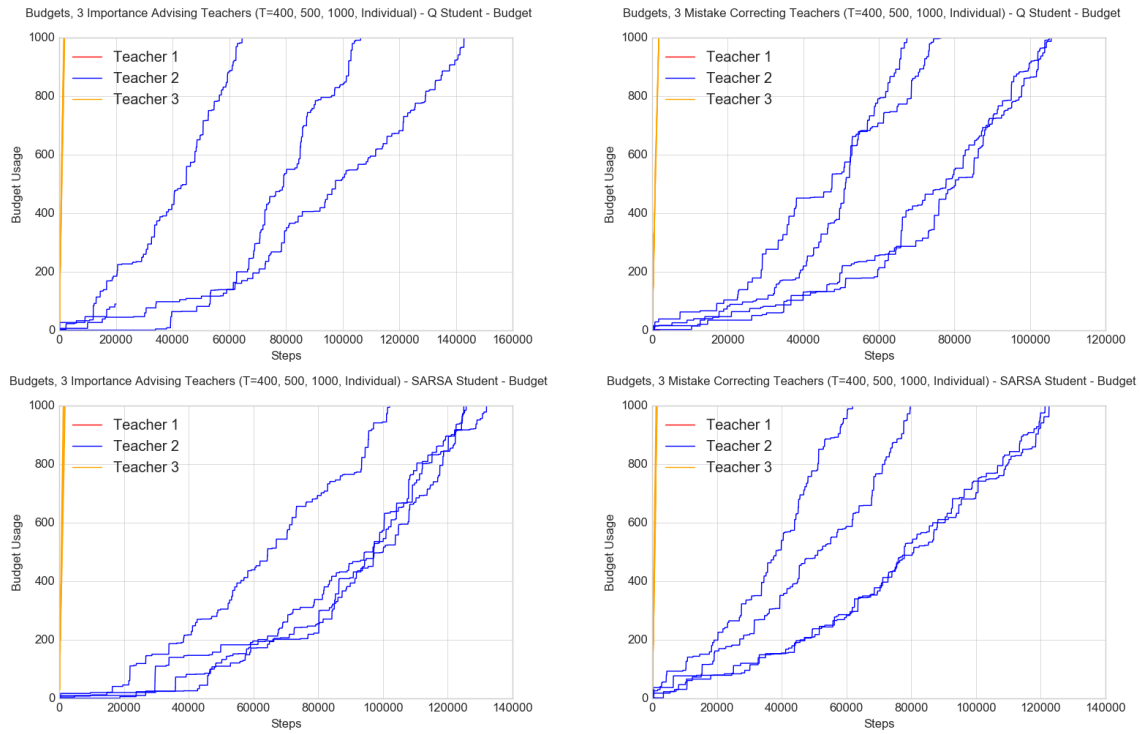


FIGURE 5.38: Comparison of 3 independent teachers' ($T=400, 500, 1000$) budget usage (Importance Advising on left and Mistake Correcting on right), for Q-student and SARSA-student

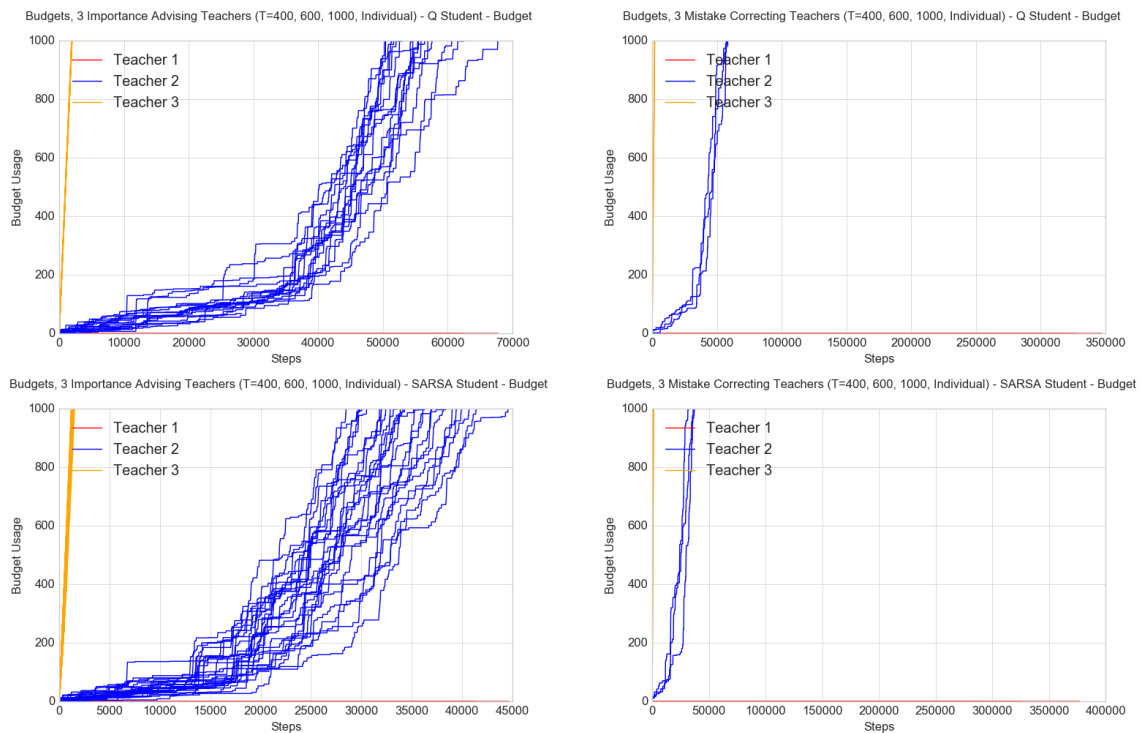


FIGURE 5.39: Comparison of 3 independent teachers' ($T=400, 600, 1000$) budget usage (Importance Advising on left and Mistake Correcting on right), for Q-student and SARSA-student

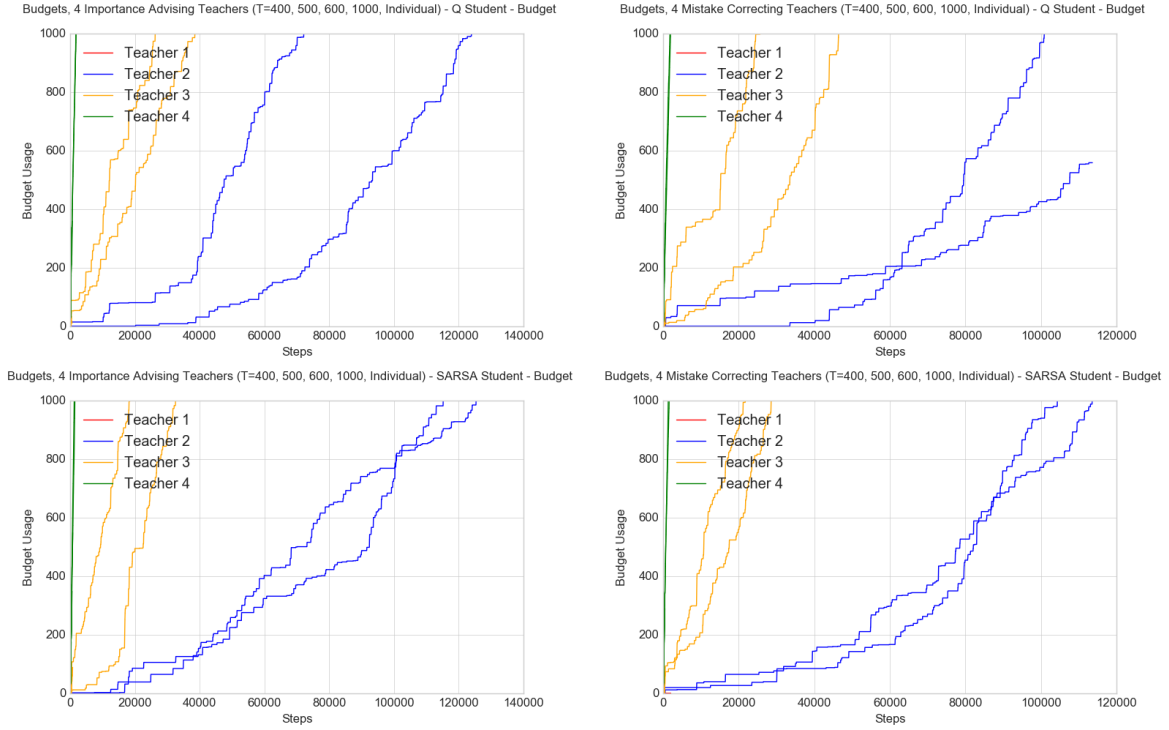


FIGURE 5.40: Comparison of 4 independent teachers' ($T=400, 500, 600, 1000$) budget usage (Importance Advising on left and Mistake Correcting on right), for Q-student and SARSA-student

the similar teachers in Table 5.1, and the p-values are shown for diverse teachers in Table 5.2. The bold p-values indicate significance at the 0.05 level; if the p-value is less than 0.05, I conclude the learning curve's area is significantly larger. I notice that only certain combinations of teachers and students in the Independent Teachers model appear to significantly perform better than the same combinations in the majority-voting Grand-Teacher model, such as five similar Importance Advising teachers teaching a Q-student. For the SARSA student, only the five Correct Important teachers yielded a significantly larger learning curve for the independent teachers; more teachers reasonably results in the utility of more individual teachers and their budgets.

# Teachers	Similar Teachers				Reversed Teachers			
	Importance		Correcting		Importance		Correcting	
	Q	SARSA	Q	SARSA	Q	SARSA	Q	SARSA
2	0.186	0.767	0.980	0.203	0.365	0.435	0.743	0.361
3	0.905	0.442	0.310	0.687	0.048	0.282	0.896	0.855
4	0.419	0.427	0.464	0.543	0.014	0.291	0.703	0.087
5	0.004	0.104	0.436	0.031	0.00006	0.118	0.562	0.871

TABLE 5.1: Significance Comparison: Independent Teachers more effective than Majority Voting (Similar Teachers)

	Diverse Teachers			
	Importance		Correcting	
Training Episodes	Q	SARSA	Q	SARSA
400	0.324	0.261	0.010	0.916
400, 500	0.725	0.143	0.663	0.299
400, 1000	0.0294	0.324	0.020	0.255
600, 1000	0.434	0.230	0.971	0.504
400, 500, 1000	0.791	0.160	0.115	0.338
400, 600, 1000	0.997	0.024	0.849	0.444
400, 500, 600, 1000	0.107	0.759	0.731	0.919

TABLE 5.2: Significance Comparison: Independent Teachers more effective than Majority Voting (Diverse Teachers)

Now, flipping the test and checking if the Majority-Voting learning curves are significantly greater in area than the Independent Teachers' curves, I obtain the p-values presented in Tables 5.3 and 5.4. Again, the p-values less than 0.05 are bold to highlight significance at that level. I notice there are fewer significantly larger areas under learning curves, but each algorithm appears to have different strengths over each other. For instance, the Q-student has a significantly larger area under the learning curve for the 400-, 600-, and 1000-episode teacher combination in Importance Advising and for the 600- and 1000-episode teacher combination in Mistake Correcting. Both algorithms merit further research.

	Similar Teachers				Reversed Teachers			
	Importance		Correcting		Importance		Correcting	
# Teachers	Q	SARSA	Q	SARSA	Q	SARSA	Q	SARSA
2	0.814	0.233	0.020	0.797	0.635	0.565	0.257	0.639
3	0.095	0.558	0.690	0.313	0.952	0.718	0.104	0.145
4	0.862	0.573	0.668	0.457	0.986	0.709	0.297	0.913
5	0.996	0.896	0.564	0.969	1.000	0.882	0.438	0.129

TABLE 5.3: Significance Comparison: Majority Voting more effective than Independent Teachers (Similar Teachers)

	Diverse Teachers			
	Importance		Correcting	
Training Episodes	Q	SARSA	Q	SARSA
400	0.676	0.739	0.990	0.084
400, 500	0.275	0.857	0.337	0.701
400, 1000	0.971	0.676	0.980	0.745
600, 1000	0.566	0.770	0.029	0.496
400, 500, 1000	0.209	0.840	0.885	0.662
400, 600, 1000	0.003	0.976	0.151	0.556
400, 500, 600, 1000	0.893	0.241	0.269	0.081

TABLE 5.4: Significance Comparison: Majority Voting more effective than Independent Teachers (Diverse Teachers)

Chapter 6

Conclusion and Future Work

This thesis confirms that the student-teacher RL framework is beneficial and has a visible impact on the student’s learning. In addition, these results appear to support the claim that multiple teachers can outperform the best teacher under certain conditions. In addition, adding “bad” teachers can hinder a student’s learning, so choice of teachers matters.

This thesis also introduces some ways budgeted teaching heuristics such as Importance Advising and Mistake Correcting can be applied to multiple teachers, namely a majority-voting Grand-Teacher framework and an Independent Teachers framework.

These results are limited, however, in that they are empirical in nature. No theoretical guarantees are made, though Zhan et al.’s theoretical Grand-Teacher model [2] is put into application in the Pac-Man domain, along with some adjustments in the algorithm. In addition, the availability of teachers and their interactions with the student (especially in the Independent Teachers algorithm) may not generalize with all domains.

I foresee several directions toward which this research can continue. First, this work focused mainly on 5 or fewer teachers, all with the same feature set. Investigating the effect of a greater number of teachers, with more varied feature sets and training times, could further enlighten the results presented in this thesis. In addition, I only used Majority Voting for choosing what action to take; trying different methods of choosing the action to advise the student (as well as when to advise) could be worthwhile. Another possible direction of research is looking into algorithms for choosing teachers, or choosing which teachers to exclude from teaching a student. This would especially be important for teachers with varied feature sets. Different budgeting techniques could also be explored, especially for the Individual Teachers algorithm.

Clearly, there is plenty of work to be done in this field of RL. Hopefully this thesis inspires more work in multi-teacher frameworks. As more complex and possibly expensive domains arise in this field, efficient teaching will be valuable for artificial intelligence.

Bibliography

- [1] Lisa Torrey and Matthew E. Taylor. Teaching on a budget: Agents advising agents in reinforcement learning. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, May 2013. 23
- [2] Yusen Zhan, Haitham Bou-Ammar, and Matthew E. Taylor. Theoretically-grounded policy advice from multiple teachers in reinforcement learning settings with applications to negative transfer. *CoRR*, abs/1604.03986, 2016. URL <http://arxiv.org/abs/1604.03986>.
- [3] Matthew E. Taylor, Nicholas Carboni, Anestis Fachantidis, Ioannis Vlahavas, and Lisa Torrey. Reinforcement learning agents providing advice in complex video games. *Connection Science*, 26(1):45–63, 2014. doi: 10.1080/09540091.2014.885279. URL <http://dx.doi.org/10.1080/09540091.2014.885279>.
- [4] Yusen Zhan, Anestis Fachantidis, Ioannis Vlahavas, and Matthew E. Taylor. Agents teaching humans in reinforcement learning tasks. In *Proceedings of the Adaptive and Learning Agents workshop (at AAMAS-14)*, May 2014.
- [5] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning : An Introduction*. Adaptive Computation and Machine Learning. A Bradford Book, 1998. ISBN 9780262193986. URL <http://ezp-prod1.hul.harvard.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=1094&site=ehost-live&scope=site>.
- [6] Peter L. Bartlett and Ambuj Tewari. REGAL: A regularization based algorithm for reinforcement learning in weakly communicating mdps. *CoRR*, abs/1205.2661, 2012. URL <http://arxiv.org/abs/1205.2661>.