Comment

More discussion of the related work is needed, in explaining the details and how they differ from the proposed work. This particularly applies to the work of Rachelson and Hoey.

There should be more discussion of the Q-PAMDP algorithm. In particular, discuss the algorithm itself (as presented in the algorithm block), the parameterisation of the algorithm which is not touched on at all at this point, and the differences for different values for k (both practically as well as what these imply for the method).

More discussion is required on the experiments. In particular, list all parameters used in the experiments. Furthermore, I would like to see more discussion on the experimental results, particularly in the differences between the two versions of the Q-PAMDP algorithm.

Make sure your figures / algorithms / tables only appear after they are referenced in the text. This applies throughout the entire document, including appendices.

Refer to Appendix A wherever it is needed throughout the document.

Mild assumptions → weak assumptions

Refer to the goal-scoring domain as the goal domain: be consistent (throughout the document)

Give forward pointers to the appropriate sections in the background introduction. Say why each piece is being discussed.

“Solving control problems with a reward signal” is a very terse description of reinforcement learning. Rather it is about learning an optimal control policy when both the transition dynamics and utility/reward function are unknown and must be learned

“receives a number indicating the cost” indicate that this is the reward, and say how the agent “receives” it.

When discussing policies, state whether you are interested in deterministic or stochastic policies in this work

“An agent interacts with an MDP by taking an action and receiving a state and a reward.” Say that this is repeatedly done, and is depicted in Fig 2.1

Fig 2.3: mention parts (a) and (b) in the caption

Fig 2.4a is very low resolution

“different choice for each state (figure 2.4a) [full stop missing]”

“A continuous state space assigns values based on a function” say that this is a function of the continuous state

In the model based vs model free discussion, make it clear that the model refers to the transition and reward functions.

In the value functions, note that the E refers to the expectation of a term under a distribution.

The discussion on how using a value function for control requires a model is not clear and needs to be expanded.

You should have a sentence explaining the TD update rule for SARSA. Also, define alpha.

The SARSA algorithm (Algorithm 1) has an empty “Input” line.

State what n is when you have (d+1)^n basis functions.

For the gradient update, comment on the relation to the value function update. Also define the gradient symbol.
More discussion on SARSA(lambda) should be given. Describe what it does and how.

Motivate why you focus particularly on policy gradient methods
“local optima theta^*” → “local optimum \theta^*”
Describe the identities that lead to eqn 2.3
Connect the equation where you introduce the baseline b, to eqn 2.3
It may be worthwhile to change the notation of the action set in the PAMDP definition from A to something like A_{d,x} to avoid confusion between a PAMDP and a standard MDP. Then use this in all future references to PAMDPs. Also formally define a PAMDP.
“A parameterized action is an discrete action” → “A parameterized action is a discrete action”
“that takes continuous set of parameters.” → “that takes a continuous set of parameters.”
“actions is totally distinct” → “actions is completely distinct”
“Fig 3.2c is low resolution
Fig 3.3 is low resolution
“jointed robot arm as” → “jointed robot arm, as”
“with respect to (\Theta)” → “with respect to \Theta”
The discussion on the policy gradient approaches should mention why this is inappropriate or “naive”.
“task parameters \tau that is fixed” → “task parameters \tau that are fixed”
Related work begins by discussing parameterised tasks – relate this to parameterised actions
“parameterized skills as a task dependent parameterized policy” → “parameterized skills as task dependent parameterized policies”
In work by Rachelson, no context on TiMDP_poly is given. How does this method relate to yours?
The work by Hoey also needs to be discussed in more depth and related to your work better

In mentioning options, mention the shortage of work in learning them, particularly as parameterized.
This chapter should start with a short recap of the problem that is being solved by the proposed algorithm. Also, make it clear that you are proposing the Q-PAMDP algorithm.
Refer to the appendix A definitions as they are used in the document.
In Defn 4.1 state that M_\theta is a discrete action MDP with a fixed parameter policy \pi_\theta
Functions W and H should be defined as referring to a particular MDP.
“updating \theta and \omega [full stop missing]”

The P-UPDATE sentence seems to come a sentence too early in the description of Q-PAMDP (just after the algorithm).
The algorithm requires significantly more discussion. Talk through the steps and what these mean for the behaviour of a learning agent. Given an example of a choice of P-UPDATE. Also, to what do the (k) and (\infty) superscripts refer? Discuss the parameter k of the algorithm, what this means, and when different values of k would be appropriate. In fact, Section 4.3 mentions the disadvantage of Q-PAMDP(\infty), but you have not even established what the advantages might be.
Mild assumptions → weak assumptions
In the proof of Thm 4.1.1, refer to it as a local maximum, rather than a local optimum, to be more consistent with the epsilon argument presented in the proof. Also state that the |\theta^* - \theta| condition is for all theta such that the epsilon bound holds.
“if we can locally optimize \theta; and” → “if we can locally optimize \theta, and”
“by selecting an appropriate \alpha” → “by selecting an appropriate learning rate \alpha”
In the statement of Thm 4.1.2: state what \alpha_t is

“difference in \omega \epsilon > 0” → “difference in \omega < \epsilon > 0”
When referring to the gradient of J with respect to \omega being zero, mention that this is when \omega = \omega^*
Summary 4.4 should also discuss the presentation of the algorithm. Also put forward pointers to the evaluation in the next two chapters.
When discussing the robot soccer domain and motivating it as an interesting domain, also make mention of the fact that it supports a number of different parameterisable actions. Also make it clear that you are controlling one robot against an adversary in this example.
“depicted in 5.1” → “depicted in Figure 5.1”

I would like to see some measure of “optimal play” in this domain as a comparison to Q-PAMDP in the results.
Figure 5.2 is not referenced in the text.
Italicise the action/skill names for easier reading.
“note that starts” → “note that the ball starts”
What is the velocity decay factor representing? This d is also overloaded with the -d reward for a terminal non-goal state.
Make it clear that the higher-level actions are designed by hand.
To what does the a_0 and a_1 refer in the kick-to command? Make this explicit.

In the experiments, it would be informative to discuss the relative times taken by the Q-learning and eNAC steps.
The last paragraph of Section 5.2 is highly redundant, twice mentioning eNAC, SARSA, and running for 50 episodes.

List all parameters used in the experiments!
“For both methods” → this is vague, mention which methods
Be consistent with capitalisation when referring to SARSA vs Sarsa

How dependent is Sarsa on different values of the fixed \theta? I would like to see how the algorithm performance varies as these change.
Comment on the differences in the results between the two versions of Q-PAMDP
When discussing the restriction in state space that only one enemy needs to be considered at a time, mention that you need only consider the current and next platform (to the right) at any point in time (I presume).
Mention in the text that the high jump is called hop, and the long jump leap
Italicise action names

Figs 6.2 and 6.3 should be swapped as they are referenced out of order in the text.
Explicitly describe the interpretation of the parameter dx for each of the three actions.
The state space is described as consiting only of four variables. What about the y position and velocity (as these describe when actions can or cannot be executed, unless action executation happens instantaneously?), as well as descriptions of the platforms? These should be part of the state space as well.
List all experimental parameters.
More discussion on the results is needed. Would optimal performance in this task not correspond to receiving a total reward of 1? State this. Why does this not happen? Also, why do the two versions of the Q-PAMDP algorithm perform differently here, unlike the previous experiment? Why does Q-PAMDP(1) show an initial dip, and Q-PAMDP(infty) have jagged points?
“One aspect of parameterized actions explored in this paper is using multiple parameterizations of a single action” → this has not been explored here. Also, this is a dissertation, not a paper!
When talking about learning parameterized skills, you don't make it clear what the future work is here.
Instead of mentioning applying action a, state that at each time step t you apply action a_t
“essentially create two policies are” → “essentially create two policies which are”
“particular strategy will be used in the goal domain” → “particular strategy is used in the goal domain”
“computational load” → “computational load”
The main equations in this thesis should be numbered, so that it is easier to discuss them in the text and also for others to refer to them
sometimes the author uses “T” to denote a matrix transpose, but sometimes this letter also refers to the number of steps in an episode. I suggest making notation more consistent;
sometimes the author uses parenthesis to denote vectors, but sometimes they use brackets for this same purpose. I suggest making notation more consistent.

Typo: “an discrete”

Typo: “takes an continuous set”

“as depicted” → “As depicted”

the last sentence of section 5.4 has a few repeated phrases and needs to be corrected

In figure 2.1, aren't the directions of the arrows reversed? The action arrow, for instance, should go from the agent to the environment,

regarding Figure 2.4, I would add a remark that the policies shown in it are deterministic policies. This is especially important since prior this figure the author was often describing policies as distributions over actions;

regarding Figure 2.5: in Figure 2.5a, arrows indicate actions (directions); in Figure 2.5b, however, arrows do not indicate an action, but the result of an action i.e., they indicate the successor state resulting from the execution of an action. This is a bit confusing.

regarding the expectations mentioned on page 11, I believe that the reader would benefit from an explicit definition of what exactly those expectations mean. Specifically, it needs to be said that E_pi is the expectation of a random variable (the sum of rewards) with respect to trajectories (sequences of states, action, and rewards) drawn from a stochastic policy pi;

the equation for Q^pi on page 11 seems to be lacking a term Q^pi(s,a) is typically defined as the total expected reward obtained if you start in state s, execute actiona, and then follow policy pi. This expectation, then (or at least the part of it related to the random variable r0), should be conditioned on action a;

on page 12 the author mentions “learning the variables a_k” These are not variables: they are coefficients, which are fixed and constant given the function that is being approximated;

on page 13 the author says that for sufficiently small alpha J(theta_{t+1}) > J(\theta_t) This only holds for determinstc policies. If there is a stochastic component to the performance, then this inequality holds only in expectation,

R(tau) is used in equation 2.2 but was never defined;

the reader would benefit from a more formal explanation of the sentence “(...) works by inverting the difference between successive theta”, on page 14;

regarding the equation for the policy gradient (page 14), I believe that the first term of its 2nd line is missing a matrix inverse operation. This inverse should appear somewhere because of the definition of the natural gradient and its dependency on the Fisher Information Matrix

Regarding Algorithm 3, please clarify that the “initial feature function” computes state features. Also, the word “initial”, here, suggests that the function itself changes, which is not the case;

Regarding Algorithm 3, the stated objective of this algorithm is to maximize the undiscounted sum of rewards, which is different from the objective that the author introduced earlier. This distinction needs to be clarified and the new objective motivated

Regarding Algorithm 3, in order for the dimensions of matrix Psi to match, I belleve that psi needs to be transposed;

What is “v”, in the last line of the algorithm? It seems like this variable is never used anywhere else. Why does it need to be estimated along with w, which is what the algorithm is really trying to compute

the author mentions “low-dimensional skills”, on page 18, but up to this point this expression has never been defined. Is a skill the same as a parameterized action? Does “low-dimensional” refer to the number of continuous numbers parameterizing a discrete action, or does it refer to the number of weights/parameters describing a policy?

if the general approach being proposed in this thesis first selects a discrete action d to execute, and the which action parameters to use, what exactly does Q(d,s) represent? Is it the total expected reward obtained by choosing the optimum continuous parameters for the discrete action d? In other words, Q(d,s) the same as max_x Q((d,x),s)?

In the first equation on page 20 (and also some other parts of the document), brackets/curly brackets are often missing. Shouldn't, for example, the expression on page 20 be {x_{i1},\ldots,...} \subseteq X_i?

The author needs to be more formal in section 3.2. What is a parameterized task? Is a task the same as an MDP? Does maximizing the objective stated on page 20 imply that the agent faces a different task/MDP at each episode? Is the probability P(tau) related to the distribution of MDPs that the agent can face during its lifetime, or is it related to the initial state distribution of a (same, fixed) MDP? If a task is an MDP, which components of this MDP can change? The reward function and the transition function? Only one of those? Or perhaps the initial state distribution of the MDP?

On page 21 the author says that options are defined as closed-loop policies that can be followed for multiple steps. Then, the author further states that “this would be useful if we want to allow for skills that are extended over a number of steps”. I wonder if there are any meaningful/useful skills which are not executed for more than one step. A discussion on this would be helpful, are parameterized actions typically short-term (or one-step) actions? If so, why?

In the first line of chapter 4, what is the “parameter-policy”? Is the author referring to the combined set of discrete policy parameters (for selecting between discrete actions) and continuous policy parameters (for selecting the continuous parameters of an action)? This expression needs to be better defined;

In chapter 4, the author uses omega to refer to the parameters of a Q function. However, omega previously referred to the parameters of pi^d. This ambiguity makes understanding chapter 4, which is a central chapter of the thesis, difficult.

The author uses (and redefines) the function “J” in too many different ways, J is defined and used in different contexts, with different types of input parameters and also denoting different quantities. A few examples: J(pi_tau, tau), J(Theta), J(theta, omega), etc. This is confusing.

On page 22 the author says: “we can compute W using a Q-learning algorithm such as (…) SARSA(lambda)” Please note that SARSA is not Q-learning. Both are examples of algorithms that are based on temporal differences, but they are different techniques.

Regarding definition 4.2 (page 22): does the variable theta, here, represent only the parameters of the policy pi^a (i.e, of the policy used for selecting action parameters)? This is how omega was being used before, but I suspect that in this particular definition, the correct variable to use would be Theta. That's because Theta (uppercase) was defined on page 20 as the pair (theta, omega), which specify the joint policy that selects both discrete actions and their continuous parameters. Using Theta would make more sense since this equation is searching for parameters omega of the Q function that accurately measure the amount of reward received by using a joint policy, parameterized by Theta.

Still regarding definition 4.2 it seems like the author is using omega for two different purposes in the text. Sometimes omega refers to the parameters of an action-parameter policy, and sometimes it refers to the parameters of a Q function. This needs to be clarified.

Regarding Algorithm 4: the author never defined what “k” (the input to algorithm Q-PAMDP) means;

in the line where theta is updated, should the expression be P-UPDATE(J_omega, Theta) instead of P-UPDATE(J_omega, theta)?

In the line where omega is updated, the algorithm executes Q-learn^infty. Does this mean that the algorithm always performs a full optimization of the Q-function with respect to the current policy? Why does Q-PAMDP allow for a varying number of steps in the execution of P-UPDATE, but not in the execution of Q-LEARN? What are the implications of this choice?

In section 4.1 the author says that Q-PAMDP “converges to [some solution] with respect to a given objective function f” Should this be J instead of f? If so, which of the many J's that were used/defined in the text?

On page 30 the author says that “to represent the discrete policy we use a polynomial basis which considers only position variables”. But isn't the policy over discrete actions implicitly defined by the soft-max distribution over Q-values, as described on page 29?

In section 5.3, how is the probability of scoring a goal defined? Is it the probability of scoring a goal in one time step? Or during an entire episode?

The graphs in chapter 5 compare Q-PAMDP(1) and Q-PAMDP(\infty), but the author did not test the proposed algorithm with other values of k. Would it make sense to also evaluate it with intermediate values of k, in order to measure the impact of running P-UPDATE repeatedly but not necessarily up to convergence? What does the author expect would happen in this case? Would the learning curves be equally-spaced interpolations between the curve of Q-PAMDP(1) and that of Q-PAMDP(infty)?

Regarding Figure 6.4, we can see that it took 140k episodes to converge. How was each of these episodes defined? Was the agent interacting, in all episodes, with a same/fixed domain (i.e. one with a same/constant set of platforms, enemies, etc) or was each episode defined as a different “game”? In the latter case, what is the probability distribution P(tau) over possible games? And if each episode is a different variation of the platform game, which aspects of the platform did the author vary when drawing a new game configuration for test in an upcoming episode?

Edit	Note
Related work has been expanded, comparisons to proposed work made.	
I've expanded on Q-PAMDP and k.	
All parameters are now listed. Discussion added on results.	
Change made	
Definitions in appendix are referenced when used	
Change made	
All references to the goal-scoring domain changed to goal domain.	
Forward pointers added.	
Explanation of reinforcement learning has been expanded on.	
Reward is now explained and how it is received added.	
Noted that we are interested in stochastic policies.	
Noted that this happens repeatedly.	
Caption now mentions both parts.	
Image resized	
Full stop added.	
Sentence rewritten	
Noted that model refers to transition and reward functions.	
Noted that this refers to an expectation.	
Expanded on discussion on why using a value function for control requires a model.	
Add explanation of TD update and alpha.	
Empty input line removed	

Added explanation on number of basis functions.
I've added a note on the relation to the value function update and defined the gradient symbol.
Expanded on SARSA lambda
Added a sentence explaining why we focus on policy gradient methods
Change made
Identities have been expanded on
Identities have been expanded on

Change made
Change made
Change made
Change made
Image resized
Image resized
Change made
Change made
Added discussion on why this approach is inappropriate
Sentence rewritten
Related tasks to parameterised actions

Change made
Rachelson's work is related to ours, context of TiMDP added.
Added discussion on Hoey
I've related options to parameterized actions and the shortage of work on parameterized options.

Recap added. Noted that the Q-PAMDP is being introduced.
Definitions in appendix are referenced when used
M_\theta is mentioned as a discrete action MDP
Functions W and H are now indexed with MDP M.
Full stop added.

Section rewritten

I've expanded on k, and the disadvantages of Q-PAMDP(inf).
Change made

All refers to optima have been changed to maxima.
Change made
Change made
Added explanation of \alpha_t

I've rephrased this slightly but the mathematics is unchanged.
Noted that gradient is zero when omega = omega*

\omega isn't necessarily less than \epsilon in magnitude.

Added the presentation and forward points to summary 4.4

I've noted the parameterisable actions and that we control one robot against an adversary.
Change made

Note: optimal play for this domain is not currently known.

Text references Figure 5.2
Skill names are now italicised for both domains.
Change made

-d reward is now \kappa. Added sentence explaining decay.
Actions are noted to be hand-designed.
Replaced a_0 and a_1 with different kick-to actions.
I've added a note on the number of episodes involved in the Q-learning and eNAC steps.

Paragraph rewritten

All parameters are now listed
Clarified which methods meant
All instances of SARSA are now capitalised.

Note: as \theta is a high-dimensional set of parameters and evaluating the performance of Sarsa for any set of parameters takes thousands of episodes, this would not be feasible.

I've discussed the differences between these algorithms.

Mentioned that we only need to consider one enemy at a time.
Actions are renamed.
Names italicised.

Referenced Figure 6.2 earlier when first introducing the domain.
dx is now described.

State space scheme has been expanded on, explanation of why we only need 4 state state variables.
All experimental parameters listed.

I've added to this section, and discussed the differences between algorithms.
This sentence has been rewritten. References to this being a paper are removed.
Added future work on learning parameterized skills.
Change made
Sentence rewritten
Change made
Change made

All equations are numbered.

Number of steps in an episode is changed from T to STEPS.

I've changed the parenthesis to square brackets.

Typo corrected.
Typo corrected.
Change made
Repeated section has been removed

Direction of the arrows has been fixed.

I've noted that the policies in figure 2.4 are deterministic.

Note: in both figures the actions represent the result of taking the action.

I've expanded on the expectation V(s).

Mentioned that r_0 is the reward received after taking action a in state s.

This sentence has been rewritten.

Noted that this holds in the expectation.
Defined R(tau)

The natural gradient is now explained in detail.

I've corrected the equation for the policy gradient.

Changed this to initial-feature functin.

Note: this computes the features for state s_0

I've noted why the sum of rewards is undiscounted.
Psi has been transposed

Rewrote last line to not use v.

This section now explains low-dimensional skills.

Note: Q(d,s) is the expected return of taking discrete action (d,x) where x is selected using the parameter policy in state s.

Brackets added.

I've rewritten the task section to clarify this.

I've rewritten this section to change this to “allow for parameterized actions that are extended over a number of steps.”

I've expanded on this.

Note: the parameter-policy selects the continuous parameters of the action.

Note: omega refers to the representation of Q. As we base pi on this Q, for simplicity we refer to omega for Q and this pi.

Note: all J's refer to the same objective function, but our agent can have different representations. This is why these parameters change.

I've rewritten this sentence.

No, we alternate learning theta and omega with the goal of optimizing Theta=(theta,omega).
Note: omega refers to the representation of Q. As we base pi on this Q, for simplicity we refer to omega for Q and this pi.

k is now defined

No, Theta = (omega, theta), we want theta.
Note: this does mean that we always perform a full optimization on Q. This is due to our theoretical basis, we don't have any results for only partially updating Q.

Changed f to J

I've rephrased this to note I meant that we represent Q(s,a) with a polynomial basis, pi is just based on Q(s,a).
Goal scoring probability is now mentioned as the probability of scoring a goal in a given episode.

Note: we only consider k=1 and k=infty because these are the values which we have theoretical results for.

Note: the domain is fixed.