Response Letter

December 25, 2018

Firstly, we would like to thank the three reviewers for their insightful comments and recommendations. We truly believe the observations pointed by the three reviewers have been very helpful to make the paper more readable.

This response letter is structured as follows. The next section summarizes the most relevant modifications addressed in the paper. Subsequent sections are devoted to thoroughly reply all the reviewers' concerns.

Main modifications addressed in the paper

We identified three main points to be addressed as per the reviewers' comments and observations. In this section, we provide details of these three main modifications, and we will point at them when replying to each of the reviewers' comments in the remainder sections. By main modifications we mean significantly large portions of new text in the paper that incorporate new explanations.

1. **Introduction**. We thoroughly revised the Introduction to make it more self-explanatory. First, we carefully revised all of the comments pointed by reviewer #2. Additionally, as indicated by reviewer #1, we provide an outline of the overall picture of our FAMA approach. Our modification is intended to adequately position our contribution within the concerned research field, and to provide a global picture of FAMA that explains our pursued objective. For this purpose, we have introduced a new paragraph that descriptively presents the functioning of FAMA. Additionally, at the end of the section we highlight the main distinctive characteristics of FAMA which describe the overall concept of our learning approach.

2. **Problem definition and complexity**. As indicated by reviewer #2, and also supported by reviewers #1 and #3, the problem definition was not clearly introduced. We believe this is the reason that our claim about using planning for solving the learning task was not fully comprehensible. We have reshaped the old section 3 (Motivation), which now is entitled ”Learning task” and contains the subsection 3.1 ”On the use of planning for solving the task”. First, we introduce the definition of the learning task (which was in section 4 in the previous version of the paper), and then we analyze the intricacies of the learning task to the different variants of observed plan traces in section 3.1. We organised the explanation

around two well differentiated cases: when the observation identifies the length of the executed plan and when it does not. Now, the discussion revolves around the particularities of solving the learning task in these two cases and we make some side comments on the complexity of the task. Hopefully, the significant remodeling of this section will clarify the need of using planning techniques for solving the learning task.

3. **Compilation scheme**. Reviewers #1 and #3 show several concerns about the understanding of section 4.4, the compilation approach. We acknowledge their concerns and we changed section 4 significantly. First, a new introductory paragraph has been added at the beginning of the section so as to provide a rough overview of the compilation approach, making explicit references to the actions that will appear in a plan that solves the planning task that results from the compilation. Second, the explanation of the set of actions that make up the planning task has substantially improved, and we provide concrete examples throughout that will help grasp the target of the three types of actions. We have also changed other minor things (e.g., the first set of actions is now renamed as **insert actions** instead of program actions) with the purpose to ease a smooth reading.

In the following, our responses to the reviewers are shown up in blue color.

Additionally, all the changes are highlighted in red in the document file.

Reviewer #1

> I suggest to describe the overall concept of the learning approach first (see above for my hopefully correct summary) and to give the technical details later, rather than describing the technical details of compilation only without providing the overall picture. Going top down would make reading much easier.

We agree with the reviewer that the paper lacks a description of the overall approach. We refer the reviewer to the main modification #1 exposed above. Following the notes of the reviewer's summary, we have also included a brief summary at the end of the section Introduction that serves to wrap up the main features of FAMA. Additionally, and as requested by the other reviewers too, we introduced the main modification #2. The new section 3 defines the learning task and provides insight of the nature of the problem to solve according to the type of the observed plan trace in section 3.1. We believe that this explanation also helps to grasp the overall picture of FAMA.

> The second reason for revision is the comparison to ARMS. I suggest to compare with LOUGA that is better than ARMS in all aspects (runtime and quality of obtained models) so ARMS is no more state-of-the-art.

As per the reviewer's recommendation, we requested the code of LOUGA to the first author of the publication entitled *LOUGA: learning planning operators using genetic algorithms*, which was published in the Pacific Rim Knowledge Acquisition

Workshop. The author of LOUGA promptly provided us with the code, and so we thank him for his gentle response. The issue we faced though with LOUGA is that this system is built as a graphical interface application, which makes it impossible to automate the model learning process and evaluation process via a script for the 154 tested cases. Nevertheless, we did manually test LOUGA on a few domains, namely the *blocksworld*, *floor-tile*, *n-puzzle* and *transport* domains. We evaluated LOUGA and FAMA in the experiment with FO action sequences and PO state trajectories with 10% of observability using 5 traces. The global precision and recall values for each domain are shown below as LOUGA/FAMA:

	Precision	Recall
blocks	0.78/1.0	0.78/0.93
floor-tile	0.83/0.75	1.0/1.0
n-puzzle	0.88/1.0	1.0/1.0
transport	0.70/0.79	0.8/0.95
AVERAGE	0.82/0.89	0.90/0.97

> Page 3, Section 2.1: The definition of effects is strange. The authors define positive and negative effects as literals. What if a negated predicate (which is a literal) is among positive effects? Does it mean the same as having the positive version of the predicate among negative effects? That is very confusing.

The reviewer is right in that the definition of effects was confusing. It all stems from the adoption of the open world assumption, which leads to states that contain positive and negative literals, but we did not adapt the definition of effects appropriately. In section 2.1, we have now included a brief explanation of the open world assumption and the representation of states, and we have adapted the definition of effects accordingly.

> Also, there are additional assumptions about the model, namely positive effects cannot be among preconditions while negative effects must be among preconditions. It would be useful to discuss these assumptions more as there are domain models that do not satisfy these assumptions. Do we really need them?

The additional assumptions are the syntactic constrains imposed by STRIPS planning models. We agree with the reviewer that other domain models do not satisfy these assumptions. We intend to extend the compilation scheme to planning models other than STRIPS as future work, but the focus of this paper is exclusively on STRIPS models. We have included a sentence before the description of the assumptions to clearly state that they derive from the syntactic constraints of STRIPS.

> Page 4: There is a nice discussion about partial traces, but I have found it incomplete because the relation between actions and state partial observability is not fully clear. For example, if we have FO for actions, does it mean FO or PO* for states? Also PO* for states looks like a special case of PO based on the definition.

I would suggest to include some discussion there.

We have reformulated the explanation about the observability in plan traces of section 2.1 to make it more comprehensible and added a few more details that help distinguish the different cases of observability. We have also highlighted the two special cases, PO* and NO, that derive from PO. With respect to the relation between actions and state partial observability, we refer the reviewer to the main modification #2. The new section 3.1 presents the combinations of observed state trajectories and observed action sequences and the impact in the learning task solving. Particularly, as it is written in the text at the end of page 10, the case FO actions with FO states is straightforward because in this case every state is available and so preconditions and effects are derivable by lifting the corresponding literals. The case FO actions with PO* states is a common case handled by some learning approaches as shown in Table 2 of section 2.2. In general, all learning approaches but FAMA assume FO action sequences, which may come along with PO*, PO or NO state trajectories.

Page 5, Table 2: I suggest including LOUGA there too.

Done. LOUGA is included in Table 2 and Table 3 and an explanatory paragraph about the main characteristics of LOUGA has been incorporated in section 2.2.

Page 10, Section 4.2: on(v1,v1) looks a bit strange. Do we really need such predicate with repeated variables?

It depends on the domain we are trying to learn. Certainly, most planning domains do not need predicates with repeated variables, but in order to ensure the completeness of our learning approach we have opted to include it.

Page 11, Section 4.3: Maybe this section fits better in the section Background. Do you use the assumption about having negative effects among preconditions and not having positive effects among preconditions like before? This is not clear from the definition of triggered.

Our approach is aimed at learning STRIPS models for classical planning problems $P = \langle F, A, I, G \rangle$ but $P_\Lambda = \langle F_\Lambda, A_\Lambda, I_\Lambda, G_\Lambda \rangle$, the task that results from the compilation, is modeled as a planning problem with conditional effects. This is the reason why conditional effects are introduced in the compilation scheme since they only intervene in the specification of P_Λ and do not condition the characteristics of the models that FAMA is able to learn. Nevertheless, we have now incorporated the explanation about conditional effects within the section 4.2 'Compilation' since this is a merely syntactic aspect of the task formalization.

On the other hand, the assumptions regarding the syntactic STRIPS constraints still apply when using conditional effects

Page 11, Section 4.4. This section is hard to understand, at least at the beginning. One needs to read it completely before understanding the details. I suggest to describe the concept at high level first and going to details then.

We refer the reviewer to the main modification #3. Section 4 has been polished significantly. An introductory paragraph provides a high level description of the approach and examples have been incorporated throughout the text.

Page 12: As I understand, the programming actions are building the model by adding preconditions and effects to the model. This immediately brings the question about symmetries as these programming actions can be used in any order (any permutation of them is fine). It would be useful to discuss this now. There are a few words about using SAT-based planner later, where parallel actions can be used. I think that this is critical for efficiency of the approach. Note also that the proposed method looks like a generate-and-test approach - we first generate a candidate model and then we verify that the model complies with plan traces. This does not seem particularly efficient.

Removing symmetries would admittedly improve performance for heuristic planners. This removal can be achieved by ensuring the actions are programmed following some order, which can be done via some additional fluents. We think, however, that the compilation scheme should remain as simple and general as possible to facilitate its understanding.

Nevertheless, we want to thank the reviewer for the great idea and we will look into adding this optimization to the approach as a compilation option.

Page 13: There is implication used in the precondition. It was was never mentioned before that a general formula can be used.

The implications can actually be encoded as a disjunction, as it is the case of the `apply_stack` action of Figure 4. The caption of the figure indicates that ”implications are coded as disjunctions”. The implication of the formula can then be encoded as $\{\neg pre_p(\xi) \vee p(\omega)\}_{\forall p \in \Psi_\xi}$.

My major problem here is that it is not clear how the actions validating states interleave with actions applying the learned actions. This must be clarified as the states are generated by actions.

The positions of the validate actions among the apply actions is determined by the planner, which will verify that the state that results after the execution of an apply action comprises the observed state of the plan trace. We have added the following sentence in the document to clarify this point: ”There will be a validate action in π_Λ for every observed state in τ. The position of the validate actions in π_Λ will be determined by the planner by checking that the state resulting after the execution of an apply action comprises the observed state $s_j \in \tau$.”

> Page 20: In the graphs, it is not clear how the values are obtained. Are these average values between the domains or mean values or is it just for one selected domain (which?). The graph with runtimes show what I was afraid about - the efficiency degrades fast with larger example plans. This raises the question if longer plans can be used if needed for some domains.

All plotted values are averages over the 14 (now 15) tested domains. The reviewer is right, however, to point out that this is not said in the experiment of section 6.2, so we have added some lines making it explicit.

Reviewer #2

> It took me a while to figure out that the authors focus on a "consistency" kind of problem, in which the task is to find a model that is consistent with the observations. This is given in section 4.1, but I would liked an organized problem definition. This is important because throughout the paper you talk about soundness, completeness and computational complexity of the problem, but this only makes sense w.r.t a clear problem definition.
>
> For example, the authors discuss in section 3 the complexity of the learning task with different assumptions, but this is done before the problem is defined. Also, all the complexity of the results mentioned in page 8 are given without a proof, and the only reference in this context is to the Russell and Norvig book. The authors write in page 8 that FAMA solves the learning problem by compiling it to a planning problem, and consequently the planning problem is P-SPACE complete, but this is not a proof. In fact, it seems like this is an incorrect reduction: if planning is P-SPACE complete and learning can be solved by planning, it does not mean that learning is P-SPACE complete, as it may be easier.

Following the reviewer's advice, we have completely remodeled section 3 and we have introduced major improvements in section 4. We refer the reviewer to our main modifications #2 and #3. We removed the complexity table of page 8, focusing now the discussion around the two main cases that determine the difficulty of the learning task and giving an indication of the task complexity according to the characteristics that define these two cases. We also changed the reference and added instead the original work that proves that boolean SAT problems are NP-complete.

The take-home lesson of section 3 is that when the observed plan trace does not provide information of the solution plan length (case 2), planning is needed to find the solution and so the learning task is PSPACE-complete.

> Repetitions.
>
> There are several cases where authors repeat themselves several times. Here are a few examples:
>
> - (page 2) "While current learning systems ... none of them allow partial observability in the sequence of executed actions " Two paragraphs later: " ... unlike most relevant action-model learning algorithms, FAMA does not require the traces

to contain any observed action ...” (Page 8) ”FAMA represents one step ahead ... without assuming observed actions.”

- A STRIPS action model is defined twice: once in section 2 and another time in section 4.

- The syntax-based notions of false positives and false negatives are defined twice: once in the second paragraph of section 5, and twice in Def. 4-5. In fact, most of page 17 seems redundant to me, and the text should be merged with that in the previous page.

We have removed repetitions as much as possible. In the first example pointed by the reviewer, the explanation about partial observability of FAMA and the other learning systems is concentrated in a single paragraph and the repetition is eliminated. Nevertheless, we think that at some points it is necessary to recall a particular concept in order to enhance the exposition of a further concept.

With respect to the repetition of the definition of a STRIPS action model, conditional effects are introduced in section 4 because these are exclusively used for modeling the task that results from the compilation, $P_\Lambda = \langle F_\Lambda, A_\Lambda, I_\Lambda, G_\Lambda \rangle$. We have now incorporated the explanation about conditional effects within the section 4.2 ’Compilation’ since this is a merely syntactic aspect of the task formalization.

With respect to the third observation, we have introduced several changes in section 5. First, we provide the standard definitions of precision and recall at the beginning of the section. Then, we adapt these two concepts to planning problems in Def. 4 and 5, respectively. Subsequently, in section 5.2, we present our novel semantic-based metrics that build upon definitions 4 and 5.

Surprisingly, adding more traces causes the precision and recall of FAMA to drop for the NO action NO states setting. The authors say this is because more timeouts were observed due to the complexity of the model learning planning problem. To support this, the authors should report the number of timeouts for the different number of traces. In particular, since we do not see this drop in precision and recall in the other settings, I expect a comparison of the number of timeouts as a function of the number of traces for both settings. If we see that the number of timeouts for this setting is larger than the simpler setting, that would support the authors conclusion.

Timeouts are only produced in the setting with NO action sequences and NO state trajectories. We have added some details on the domains where timeouts are found: in the grid domain when $|\mathcal{T}| = 3$, floortile when $|\mathcal{T}| = 4$ and four other domains. We refer the reviewer to the comment added in page 23 of section 6.2.

It was a bit weird to me that the authors chose to compile to a planning and not to a SAT problem like many prior work on learning action models.

One reason to compile to planning is when the length of the solution plan is unknown, but the authors say that this is known in this case as well (see my question on this later).

From my understanding of this work, there’s a variable for every tuple of ac-

> tion name, fluent, and precondition/add-effect/delete-effect option. Then, there are constraints to hold that all this fall together in a consistent way with the observed trajectories.
>
> So, my question is: can the authors motivate the choice of planning over SAT?
>
> Since planning problems can be solved by compilation to SAT, and SAT problems can be solved by compilation to planning, it does not really matter to which problem you compile to, but still, I wonder what motivated the authors to take this approach.

The reviewer is totally right in that planning is needed when the length of the solution plan is unknown. Precisely, this is the point that motivates our approach but that we did not successfully convey in the previous version of the paper. When the observed plan trace comprises a partially observed action sequence and a partially observed state trajectory, we ignore the length of the solution plan and so we need to resort to planning. The new section 3 explains this in detail (main modification #2). When the length of the plan is known, SAT-solvers can be used; when it is not known, planning is required. In our case, we use a planner to solve all learning tasks independently of their nature.

On the other hand, we happen to use a SAT-based planner (Madagascar) for solving the learning tasks, which may lead to wrongly interpret that the length of the solution plan is always known. The reasons for using Madagascar have been highlighted in red at the beginning of section 6. In a nutshell, we use Madagascar because it returned the best results compared to other planners and because of its ability to handle problems with dead-ends. Additionally, Madagascar is obviously very suitable to solve learning tasks when the length of the plan is known.

> Page 1. In general, the introduction left me very confusing, and I only really understood it after reading the paper. I wish the authors could do a better job in having the introduction not rely on understanding the rest of the paper so much. Concretely, talking about FAMA as a ”solving scheme” ”whose solution must be compliant with the input plan traces” did not make sense to me in the first pass.

We acknowledge the reviewer’s concern and refer the reviewer to the main modification #1 exposed at the beginning of this response letter.

> Page 1. ” an abstract version of the capability model that reflects ... ” – I am not sure that ”abstract” is the right term here, since the learned model is not an abstraction of the real-world but an approximation of it.

We have replaced the word ’abstract’ by ’approximation’ and rephrased the sentence a little bit.

> Page 1. ”Two different types of data are generally identified in the generation of explanations (2) data that do not accurately represent the decision-making process because the observed external behavior of the agent responds to a blackbox model.” – First, I think you meant ”corresponds” and not ”responds”. Second,

if the agent is treated as a black box, then this is not really a way to generate explanation. Please better explain the text after (2) .

We think this paragraph was rather confusing and that we did not clearly come through our intention to set an analogy between the data sources used in explainable AI and the data required by the learning approaches. For this reason, we have removed the paragraph and included instead a list of planning-related tasks that require a planning model for being solved. We think this rather complete but non-exhaustive list is a pretty good indication of the need for learning planning models.

Page 2. " ... the synthesis of different kinds of generative models with classical planning ..." – I do not understand this sentence. What do you mean by generative models in this context?

Page 2. "FAMA is a solving scheme ..." – FAMA is a learning scheme, not a solving scheme. I agree that it is a learning scheme that learns by solving a planning problem.

Page 2. " ... by the input observed actions, ... " What is "input observed actions"? please rephrase.

We have rephrased all the paragraphs and sentences pointed by the reviewer. With respect to the first comment, we attempt to make it clear that classical planning techniques can be used to generate a large variety of planning models, not only STRIPS models, and that these techniques underpin some relevant concepts that are also reusable in learning models when the behaviour of the agent is not available.

With respect to the other two comments, besides using different expressions to refer to the two concepts, we have included some additional side explanations that we hope help understand the meaning of the sentences.

Page 3. " ... the open world assumption" – Either explain what this means or provide a reference.

We have included an explanation of the open world assumption and how this affects the representation of states in Section 2.1.

Page 3. " ... the minimal action sequence to transit from state s_i to s_{i+1} is composed of a single action ..." - This can be stated in a simpler way: "there exists a single action that transitions s_i to s_{i+1}"

Fixed.

Page 4. "A plan trace .... For a planning frame ... holds that ... for every action in ... and that ..." - this sentence is not clear. Please rephrase.

The sentence was, indeed, unclear. Anyway, we removed it in the revised version of the paper since it was an expendable phrase that did not fit very well in this section either.

Page 4. The term action header is not defined.

We have moved the 4th paragraph of the Related Work, where we commented on the predicates and action headers of the approaches, to the end of the section since it makes more sense to comment on this issue when comparing with LOCM. Also, we clarified the meaning of action header (action name and parameters) because the formal definition of an action header is actually introduced in section 4.

Page 4. " ... of the evaluation method to validate ..." - missing "used" before "to"

Fixed.

Page 4."Exceptionally, a NO state sequence in LOCM is a fully-empty trajectory, with neither initial or final state. " - I don't understand this sentence. Isn't a NO state sequence a fully empty trajectory for all algorithms? Also, it should be "nor" and not "or".

The labels FO, PO*, PO and NO concern exclusively the intermediate states of the state trajectories as indicated in Table 1. We have improved the explanation of the four types of state trajectories in section 2.1. As for LOCM, this approach does not require any state, not even the initial or final state. We changed the label of LOCM in the first column of Table 2 to "—" to indicate that no state information at all is needed. We have also rephrased the sentence about LOCM.

Page 4."Most approaches assume that a set of predicates and a set of action headers are provided alongside the input traces" - What about FAMA? I think it also requires this.

The reviewer is right in that FAMA also assumes that predicates and action headers are provided or otherwise are extractable from the plan traces. We believe it is more appropriate to move this comment after LOCM is introduced since LOCM is an exception that learns the predicates using Finite-State Machines (FSM) because no observed states are given in the plan trace.

Pages 5-7. When describing prior work, it is sometimes not clear when you describe the algorithm and when you describe their specific implementation and experiments. Specifically:

1. When describing SLAF, you write that 1,000 action-observation sequences and 10 fluents are selected. Is this part of the algorithm or just how they specifically implemented it in their experiments?

2. When talking about LAMP, you write that it allows trajectories up to a "minimum percentage of 1/5 " again, is this requirement mandatory in their algorithm, or is it just what they happened to get in their experiments? Also, if you write 1/5 then you should write "ratio" and not "percentage".

3. The same question applies to your description of CAMA, when talking

about 80% empty states.

EVA ...

Pages 5-7. " ... the state of a sort" – " what is a "sort"?

EVA ...

Pages 5-7. "As many of the approaches in section 2 assume, there may be an ..." - this text is redundant. It just repeats again what was said twice already.

We refer the reviewer to the main modification #2. Section 3 has substantially changed and this sentence does no longer appear in the text.

Page 8." ... an unbound number of missing actions ..." - should be "unbounded'

Corrected.

Page 8.I disagree that having FO implies having human observers annotating the traces. For example, if one monitors a robot performing actions by monitoring the communications between the robot and its controller.

It is true that a machinery for capturing and labeling the executed actions can be deployed. In either case, whether a human or an automated mechanism, the need of a fully observable behaviour of the agent imposes a limitation in learning systems. We have soften the sentence including the word "*commonly*" in our exposition.

Page 8."When the plan traces is fully observed, learning STRIPS action models is straightforward" - The reference if for work done in 2018, while learning an action model in this setting has been discussed in IJCAI 2017 by Stern and Juba (Safe, model-free learning of action models").

SERGIO ...

Page 8.The use of syntax and semantics in this stage was confusing to me. I am not sure how to improve this, though.

We have removed the paragraph regarding the evaluation of the learned models from section 3 and moved to section 5. We also clarify the use of the word "semantic" as requested by another reviewer as well.

- "On the other hand, a striking figure ..." - I could not understand this sentence.

In the remodeling of section 3, the paragraph about the size of the training dataset has been removed as well. The ideas presented here have been moved to page 23, at the end of section 6.2.

- ”Let $\Omega_v = {v_i}_{i=1}^{max_{a \in A} ar(a)}$ be a new set of objects ” I think ”$ar(a)$” should be ”$|ar(a)|$”

The function $ar(\cdot)$ does not return a set, it returns the arity of an action or predicate; i.e., the length of its argument list. This is introduced in the Background section for the first time.

- ”... which is bound to ...” - I think should be ” which is bounded by”

Corrected.

- ”In more details, for a given ...” - delete the ”In more details” as you are defining something new, not providing additional details.

Corrected.

- Please elaborate on the computation of the size of the space of possible STRIPS models.

SERGIO ...

- The definition of conditional effects given in 4.3 should be moved to the background section, where you define a STRIPS model.

Our approach is aimed at learning STRIPS models for classical planning problems $P = \langle F, A, I, G\rangle$ but $P_\Lambda = \langle F_\Lambda, A_\Lambda, I_\Lambda, G_\Lambda\rangle$, the task that results from the compilation, is modeled as a planning problem with conditional effects. This is the reason why conditional effects are introduced in the compilation scheme since they only intervene in the specification of P_Λ and do not condition the characteristics of the models that FAMA is able to learn. Nevertheless, we have now incorporated the explanation about conditional effects within the section 4.2 ’Compilation’ since this is a merely syntactic aspect of the task formalization.

- The explanation about the plan prefix and postfix is confusing and not helpful. I recommend removing it. Only after you explain your solution it starts to make sense. Also, I am not sure the term ”postfix” is suitable. Maybe ”suffix”?

We have introduced a new paragraph at the beginning of section 4 that gives an overview of the compilation approach and explains the two blocks of actions of the solution plan to the learning task. We have reused the concept of ”block of actions” to refer to the prefix and postfix of π_Λ in section 4.2 ”Compilation”.

- The formula for the precondition of apply uses the notation $\Rightarrow$ which was not defined.

$\Rightarrow$ is the general logic symbol read as implication. The implications can actually be encoded as a disjunction, as it is the case of the `apply_stack` action of Figure 4. The caption of the figure indicates that ”implications are coded as disjunctions”. The implication of the formula can then be encoded as $\{\neg pre_p(\xi) \vee p(\omega)\}_{\forall p \in \Psi_\xi}$

Pages 15-19. - I'd appreciate more details on the mentioned post-processing, e.g., a pseudo code.

The post-processing is applied to learning preconditions that are static literals and are never deleted or added. We inherit the same procedure for learning preconditions presented in [40]. The pseudo-code is presented in [40] (algorithm 2).

PENDIENTE DIEGO MIRE UNA FRASE QUE LE HE MANDADO POR EMAIL

- At the bottom of the page, you write that VAL cannot address the model validation of a partial model and on the other hand it requires a full plan and full action model. I don't see how this is "on the other hand"? it is the same "hand"

Corrected.

Pages 15-19.

- When introducing the semantic evaluation metrics, you provide two justification. One is that there is no GTM. The other is that a test-based evaluation is preferable. The latter justification is not really a justification. It is like saying "we justify using a test-based evaluation by the fact that test-based evaluation is sometimes preferred".

We have provided better justification for the use of semantic evaluation metrics in section 5.

Pages 15-19. - What does it mean that a learned action model is sound and complete?

The precision and recall metrics give a notion of the soundness and completeness of a model. By a sound model we understand one where all preconditions and effects are correct, either wrt to the GTM or a test set, so there is no need to remove anything. A complete model is one where no precondition or effect is missing. We have added a sentence to clarify this.

Pages 15-19.

- "... to flaws that appear more than once in the plan traces ..." - I recommend to clarify: "flaws in the action model that manifest more than once in the plan traces".

- "... generated 10 plan traces (each with 10 actions ..." - the parenthesis should be removed.

Fixed.

Pages 15-19. - "... the horizon of the solution plan is also known. " - how do you know the horizon? You know the length of the given traces, but how do you know how many actions are needed to build the action model?

Hopefully the new section 3 (see main modification #2) helps clarify this question. In a nutshell, sometimes a plan trace indicates the length of the plan (this is the case when SAT solvers can be used) and others it does not (this is the case when planning is required).

- ”... a solution plan is solvable in two steps” - which two steps? I thought you put it all in the planner and get both action model and validation in the same step.

Yes, the reviewer is right in that we put it all in the planner and get the overall plan, both building actions and validating actions. What we mean by ”solvable in two steps” is that a SAT-based planner internally solves the construction of the model in two *time steps*: in one time step the truth value of all the model fluents that represent the preconditions is set, and in a second time step the value of all the model fluents that represent the effects is set. We have slightly modified the sentence to make it clear we refer to the SAT solving process of Madagascar.

Pages 20-end of paper:

- ”... the size of the input knowledge, in the performance ...” - should be ”on the performance”
- ”... the Precision and Recall” - why capitalized?
- ”By lightening the input constraints ...” - maybe ”relaxing” instead of ”lightening”?

Fixed.

- ”retain a level of soundness similar to ” ... what is this ”level of soundness”? do you talk about precision?

Yes, precision gives a notion of soundness since it measures how many of the learned preconditions and effects are correct. We have improved the explanation of the metrics in section 5 to make this more clear.

Reviewer #3

First sentence of the Introduction: there is no satisfactory or generally accepted definition of ”complete” domain model, given that a domain model is an abstraction of the domain being modelled. If the authors use such a term they should make clear what their definition is. ”Adequate” is a better word to use - in the sense of being adequate enough for use with a planner to solve a set of planning tasks.

We have replaced the word ”complete” by ”adequate” as suggested by the reviewer.

In this part and the rest of the paper I do not fully agree with the author's hype about FAMA with respect to the input information required. The minimal amount of information that is needed (if I have got this right) is to specify every predicate that will be needed in the application (this is done in the initial state) the number of steps that are in the training plan, the set of all action headers including their list of parameters, and a goal state which refers to the same predicates as the initial state.

Comparing this to, say, LOCM, where no state/predicate information at all needs to be specified, I would say that in fact LOCM requires a lot less 'information' a priori. What FAMA does not require is the identity of all the actions at each step, nor the instantiation of their headers, but it does require the definition of all the predicates that may appear in any state. The truth it seems to me is that FAMA would be good for some applications which fit this set of constraints, which is a different set of constraints than for other domain model learners.

A way forward might be to put the claims about FAMA into a smaller set of learning systems (certainly ARMS will be in that set) where it clearly needs less input information.

We agree with the reviewer's observation, and we believe the misunderstanding stems from us mixing up the words *information* and *observability*. In our claim of FAMA using the minimal amount of information, we should actually say that FAMA uses the *minimal observability* possible, being the extreme case to have no actions at all and only the initial state and final state as observations. In this respect, LOCM is indeed the learning approach that requires the *minimal information* since no state or predicate is needed.

We think that LOCM is a learning approach that notably differs from the rest of systems presented in Table 2 and, in this regard, we have highlighted this distinction in the explanation of section 2.2. Besides adding a sentence that stresses the ability of LOCM to learning with minimal input knowledge, we devote the last paragraph of section 2.2. to explain the distinctive features of the LOCM family.

NO ESTA HECHO TODAVIA

I totally agree with the authors in their assertion that the evaluation methods for previous model learners is currently poor, and amount to how close the learned model is to a GTM. I am not convinced, however, that the use of the statistic methods proposed is a great improvement. Precision and Recall are heavily used in data mining where (generally) there are huge data sets and there is not generally a hand crafted model for comparison. In planning, however, the idea of separating out a set of traces into training and testing appears odd, especially where it is known that the learning will eventually create a model that explains all the training traces. This seems particularly odd if there are only a few traces to be split up (and FAMA appears only able to cope with a few traces because of complexity problems).

The syntactic metrics commonly used by other approaches count the number of differences wrt the GTM and normalize by the total number of possible preconditions and effects. This gives rise to an optimistic measure of the error since the set of possible preconditions and effects is usually much larger than the size of the actual GTM, resulting in low error rates even when the model is largely flawed. In contrast, precision and recall normalize by the size of the GTM, thus removing this problem. Furthermore, precision and recall provide an intuitive notion of soundness and completeness, respectively; whereas a sound model is interpreted as a model in which all preconditions and effects are correct wrt the GTM, a complete model is one in which no precondition or effect is missing. We have added some paragraphs to section 5 to better explain these concepts.

As the reviewer points out, FAMA will create a model that explains all the training traces because it is a satisfying approach (that satisfies all the training samples) and so the training error is 0. There is, however, no guarantee that the training traces are able to capture the whole dynamics of the domain. This is specially true for scenarios with low observability like the ones explored in this paper. Hence, it is always interesting to measure how well the learned model is able to explain a new set of plan traces.

> - The method of learning: This is to compile the learning problem into a conditional-operater classical planning problem. This method appears to be highly innovative to this line of research (stemming from an earlier ICAPS paper by the authors).
>
> The description of the compilation method did not appear to me to be at the level of the detail to allow easy reading. It took me a good deal of time to work out what going on. This is a complex process and I recommend much more use of examples throughout - perhaps a running example to show the full run of a compilation.
>
> It might also improve the paper if an algorithmic description of compilation were included? One aspect that was not obvious from the description was how a SET of models are delivered by planning rather than just one; this needs spelling out.

We have substantially changed the explanation of the compilation scheme. We refer the reviewer to the main modification #3. Besides an introductory paragraph that provides an overview of the compilation, we have included examples throughout the text to clarify the role of the insert actions, the apply actions and the validate actions. We have also renamed some concepts with more illustrative names.

> - The evaluations of the domain model using classification metrics of precision and recall: I like the distinction of the methods labelled ”syntactic” (using the GTM) vs ”semantic” here; however I think the use of the word - ”semantic” a bit misleading. This implies taking into account the meaning of the features within the application domain which are left out or in erroneously compared to the GTM e.g. that a missing ”on(x,y)” in a learned model is somewhat better/worse from a missing holding(x) - because of their meaning, rather than counting each predicate as ’1’.
>
> In the end your method reverts to counting the number of different predicates, but in the context of the recall/precision formula with respect to domain model(s) that are consistent with the test sets, not only the GTM.

The definition of the metrics was not too clear. Intuitively, I think I understood 5.1 but it was not well stated -the definition of some terms seemed syntactically wrong or at least sloppy.

E.g. in the definition of INS and DEL, should not the universally quantified variables be further qualified by stating that there is a union of the already unioned sets? Or again, should there not be an extra ”Sigma” to represent the addition of the addition of the number of precons and effects in size(M) definition? In proposition 7 you say the closest set ”is the GTM” do you mean equal to the singleton set of the GTM?

Also - the structure of 5.1 - it would be best to clearly state that the syntactic evaluation is going to be introduced, then as a further subsection the semantic evaluation.

The semantic evaluation is used to measure the ability of a learned model to produce a given plan trace. In this context, the use of the word ”semantic” refers to the physics of the domain, not the actual meaning of the predicates or actions. We have added a comment clarifying this point.

We agree with the reviewers concerns about the definitions of INS, DEL and size(M) and have corrected them.

Following the reviewer recommendation, we have splitted section 5.1 into two sections: one for the syntactic evaluation and another for the semantic evaluation

- I advise changing title Experimental Results => Experimental Evaluation

We changed the title as recommended by the reviewer.

- At the beginning of the evaluation you state the obvious - that the purpose is to evaluate the performance of FAMA and the quality of the models. At this stage to clarify the experimental section I suggest that you indicate, before detailing the set up, what would be success, for example how do you intend to show that your technique is better than or comparable to others? This will no doubt mean introducing the metrics you intend to use and giving an idea of success in terms of them. Also it may mean taking text from the start of the sections 6.2 - 6.5 into the first part of the evaluation section. This will however give a good rational for the ’set-up’, the choice of the particular experimental scenarios, and for the discussion of the results.

We have introduced a paragraph at the beginning of section 6 that gives an overview of the experiments presented in this section. The quality of the models (*success* in the words of the reviewer) of the three first experiments is measured by the syntactic metrics introduced in the evaluation section; that is, the quality of the models is compared against the corresponding GTM. The last section presents a syntatic vs semantic evaluation of the learned models. The semantic evaluation measures the precision and recall of the models to reproduce the input plan traces. All this is explained in the introductory paragraph. With respect to moving text from the subsections of section 6 to section 5, we think the text of these subsections is particularly contextualized to the experiment in question and seems better to leave the text in section 6. Nevertheless, we have improved some explanations in section 6.

> - Why ”14” models? please give some explanation (e.g. there are only 14 strips models there) so we know why/if you have left any out

We keep adding new domains as we work on the approach so this number is representative of how many domains we have tested our approach on. We did, however, leave one domain out, *Sokoban*. As a matter of fact, this domain presents some of the most difficult features to be able to be handled by our approach, particularly features intrinsic in puzzle-like domains and domains with grid-like structure. We have though included one more domain, the *rovers* domain.

> was the max predicate arity of ”2” in all of the 14 a big issue? Given the inherent computational problems in planning I guess so?

Predicates with arity 3 are certainly harder but still manageable. In order to prove this, we have added the *rovers* domain to the list of domains tested in section 6.

> so for the NO / NO case study, we know the full IS , GS, the number of action steps, and the space of possible action headers for each step - is that correct? If so please make it more explicit - it not say explicitly what it means. This was not very explicit in the introduction/compilation process description either.

> Typos spotted
>
> ”around a hundred of traces”p.20
>
> Please replace ”fairly sound and almost totally complete” with something more precise. p.21
>
> ”reason why the scores of” p.26

Fixed.