Leveraging Neurosymbolic AI for Slice Discovery

Michele Collevati[a], Thomas Eiter[a], and Nelson Higuera[a]

[a]Institute of Logic and Computation, Technische Universität Wien
michele.collevati@tuwien.ac.at, thomas.eiter@tuwien.ac.at, nelson.ruiz@tuwien.ac.at

1 Introduction

We thank the reviewers for their careful reading of the manuscript and the suggestions to improve this work. In reaction to the comments, we have incorporated the minor corrections and furthermore substantially enriched the manuscript with further material, in particular with (1) more data concerning the experiments on the *Super-CLEVR* dataset and (2) a new Computer Vision task, viz. image classification, besides object detection, which is considered over a real-world image dataset based on the popular *ImageNet* dataset. For the new vision task, extensive experiments have been conducted and data collected similarly as for the previous task. The results obtained confirm the viability and usefulness of the approach.

The paper has significantly gained from the revision, which due to the requests of the reviewers necessarily incurred an amount of work that did not make it possible to stick to the original deadline for the revision. We are confident, however, that the paper now meets the requirements as the issues of the reviewers have been addressed. In addition to the requested changes, minor corrections and linguistic improvements have been made. In order to ease the work of the reviewers, relevant changes in the document are shown in blue (excepting headings and tables that were difficult to color).

In the sections below, we respond in detail to the comments of the reviewers.

2 Review 1

1. The technical contribution is limited. Further experiments would be needed to support the general applicability of the proposed task and to compare it against the state-of-the-art.

   `Answer:` To better support the general applicability of the proposed Slice Discovery Method (SDM), we have significantly extended our experimental evaluation in the revised manuscript in Section 6 ("Experiments") and Appendices C and D. Specifically:

   (a) We have added new experiments on the real-world image dataset *ImageNet* to validate our SDM on an image classification task. This complements our original experiments on the synthetic *Super-CLEVR* dataset, which focused on object detection, and demonstrates the versatility of the SDM approach across different Computer Vision (CV) tasks and its effectiveness in a more challenging and realistic domain.

(b) We have included more iterations of our SDM pipeline for the experiments. This better illustrates the iterative nature of the slice discovery and mending process and its ability to handle increasingly subtle model failures after an initial repair.

We believe that these substantial additions, which now cover different tasks and datasets, better support the utility and general applicability of our method. Our extended experimental evaluation now more robustly supports the claims made in this paper. Furthermore, while a direct performance comparison is not applicable in our context because other systems do not produce symbolic rules, in Sections 2 and 7 ("Related Work" and "Discussion") we technically compare our work with state-of-the-art methods, highlighting the advantage of our neurosymbolic SDM in addressing the fundamental challenge of slice discovery interpretability.

2. The manuscript only considers one dataset (Super-CLEVR) and one task (object detection), which is relatively simple and does not include object relations as well as properties. More tasks should have been included in the extension to demonstrate how the proposed methodology could be applied.

   `Answer:` Based on this suggestion, as described in Point 1, we have added new experiments on the *ImageNet* dataset to validate our SDM on the image classification task. This demonstrates that the SDM approach is not limited to a single domain but can be effectively applied across different CV tasks and datasets, as described in Section 6 ("Experiments") and Appendix C and D. Regarding object properties, our SDM is fundamentally based on them. As detailed in Section 4.4 ("Rule Extraction via Inductive Logic Programming"), our SDM approach converts image scene graphs into logical representations that explicitly encode attributes for each object, such as its shape, colour, material, size, and direction. These attributes are crucial for discovering rare slices defined by these specific object characteristics. Regarding object relations, it is correct that the current work focuses on discovering rare slices defined only by the object attributes. As we now state in Section 8 ("Conclusion and Future Work"), incorporating relationships between objects (e.g., discovering a slice like "a bicycle next to a car") is a key direction for our future research. This extension will allow for the discovery of more specific rare slices subject to increased difficulty for parsing and learning.

3. The proposed methodology requires complete scene graph descriptions, even though the task is considerably simpler (object detection). In its current form, it assumes that additional labels, beyond those needed for the task at hand, are available.

   `Answer:` We agree that our SDM relies on scene graph descriptions, which contain more information than the target class labels used for training a *YOLOv5* object detector. This is a fundamental aspect of the neurosymbolic SDM approach, as these detailed attributes are precisely what enable Inductive Logic Programming (ILP) systems to discover and describe rare slices. To demonstrate the feasibility of our SDM, the experiments were designed to cover two distinct scenarios:

   (a) For the *Super-CLEVR* experiments, we used the ground-truth scene graphs provided by the generator. This allowed us to validate the SDM approach in a controlled setting, proving its effectiveness when a perfect semantic description is available.

   (b) For the *ImageNet* experiments, since real-world datasets do not come with ground-truth scene graphs, we demonstrated a more realistic setting by using *GPT-4.1* as the Vision

Language Model (VLM) for generating the necessary scene graph descriptions directly from the images.

Our SDM is based on image scene graphs, but it does not assume that they are already available for a dataset. We have clarified this in Sections 4.3 and 7.4 ("Scene Graph Generation" and "Limitations"), and highlighted the scene graph generation via VLMs as a promising research direction in Section 8 ("Conclusion and Future Work").

4. The resulting slices are dependent on the chosen ontology. Further details would be beneficial for the reader to understand how the ontologies were constructed and why the original Super-CLEVR ontology was not used. Given that the selected ontology constrains the rare slices that can be discovered and generated, I believe this aspect is important.

   `Answer:` We agree that the resulting slices are dependent on the chosen ontology. This is an intentional and important feature of our methodology. To test our neurosymbolic SDM in identifying rare slices, we first had to induce systematic errors in CV models in a controlled manner. To achieve this, we applied a taxonomy-based heuristic that separates visually similar vehicle subclasses into different target classes. Regarding the original *Super-CLEVR* ontology, we want to clarify that our experimental evaluation did include it as the *VT:$\mathcal{H}3$* hierarchy. We have clarified this point in the "Taxonomies" paragraph within Section 6.2.1 ("Experimental Setup"). However, to create more challenging scenarios to test our SDM, we also designed other custom hierarchies (e.g., *VT:$\mathcal{H}4$* and *PP:$\mathcal{H}1$*) that separate a greater number of visually similar vehicle subclasses. We applied the same principle to the *ImageNet* experiments. Instead of using the entire, vast WordNet hierarchy, we created a custom taxonomy from a selected subset of vehicles from WordNet, again with the goal of inducing the generation of rare slices in the realistic domain of *ImageNet*. This heuristic-driven approach to taxonomy design is fundamental to our methodology, as it allows us to create a controlled and reproducible testbed for evaluating SDMs. We have clarified this point in the revised manuscript in the "Taxonomies" paragraph within Section 6.2.1 ("Experimental Setup") and in the "Taxonomy" paragraph within Section 6.3.1 ("Experimental Setup").

5. The proposed methodology was not compared against any baseline apart from the original model prior to mending. However, multiple techniques have been proposed in literature to discover and tackle rare issues, both in the field of SDM and in the more general field of active learning and long-tail learning. It would be appropriate to compare the proposed methodology against other SDM methods, at least in terms of technical characteristics, if not in terms of performance. Alternatively, the effectiveness of the proposed methodology in mending the model could be compared against simple baseline, e.g., taken from the active learning literature (as done in Jiang, Chiyu Max, et al. "Improving the intra-class long-tail in 3d detection via rare example mining." European Conference on Computer Vision. Cham: Springer Nature Switzerland, 2022).

   `Answer:` In both Section 2 and 7 ("Related Work" and "Discussion"), we now address this point by technically comparing our SDM approach against recent methods such as *Domino*, Talisman, and the work by Jiang, Chiyu Max et al. (2022). These methods typically operate in the embedding space to identify underperforming data regions. In contrast, our framework prioritises interpretability by leveraging ILP to extract human-readable logical rules identifying rare slices.

6. Tables 1, 2, and 3 offer a detailed comparison of the three ILP systems used, but it is not so straightforward to compare. A few summary measurements could be useful. Also, the number and quality of the generated rules should be compared.

   `Answer:` In the revised manuscript in Sections 6.2.2 and 6.3.2 ("Experimental Results") and Appendix C and D, we have now introduced new summary tables (e.g., Table 5, 6, etc.) that aggregate the performance of each ILP system across all runs for each hierarchy. These tables provide a clear comparison based on specific summary measurements, i.e. total runtime, total number of rules, and total number of rules per vehicle subclass (from which we infer the quality of the generated rules).

7. It is not clear which ILP configuration was used to generate the rules for model mending, and how sensitive the mending process is to the specific choice of rules.

   `Answer:` We clarify that our methodology does not rely on selecting rules from a single, arbitrary ILP configuration. Instead, our approach is more robust. The candidate rules used for model mending are derived from a consensus analysis across the entire set of extracted rules from all ILP systems and configurations. We select the attributes (e.g., the `utility_bike` vehicle subclass and the `north` direction) that appear most frequently and consistently. This ensures that the rules guiding the model mending are stable and not just an artefact of a specific hyperparameter choice. We have made this process clearer in the "Rule Extraction and Selection" paragraph within Section 6.2.1 ("Experimental Setup"). While our experiments confirm that model mending based on our consensus-driven rules is effective, a full sensitivity analysis represents a significant research effort that is beyond the scope of this current work. We agree that a systematic study of how the choice of rules impacts model mending effectiveness is an important research question. Therefore, as suggested, we have highlighted this as a future work in Section 8 ("Conclusion").

8. Experimental settings for training the YOLOv5 model should be given for greater reproducibility.

   `Answer:` We have revised the manuscript to include the specific experimental settings used for training the *YOLOv5* models in both experiments. The following details have now been added to the "Neural Network" paragraph within Sections 6.2.1 and 6.3.1 ("Experimental Setup") of each respective experiment:

   - *Super-CLEVR* (object detection) experiments: For each of the five hierarchies, a *YOLOv5* model version *yolov5s* was built on the training set running 80, 160, and 320 epochs using an image size of 640×640 pixels and a batch size of 16. We used the default *YOLOv5* hyperparameters provided by the official implementation, which include the *SGD* optimiser, initial learning rate of 0.01, final learning rate factor of 0.01, momentum of 0.937, and weight decay of 5.0×10^{-4}.
   - *ImageNet* (image classification) experiments: For the *VE:*$\mathcal{H}$*1* hierarchy, a *YOLOv5* model version *yolov5s-cls* was built on the training set running 20, 40, and 80 epochs using an image size of 224×224 pixels and a batch size of 16. We used the default *YOLOv5* hyperparameters provided by the official implementation, which include the *Adam* optimiser, initial learning rate of 0.001, final learning rate factor of 0.01, momentum of 0.9, and weight decay of 5.0×10^{-5}.

We believe this addition makes the experimental setup much clearer and enhances the reproducibility of our work.

9. To evaluate model mending, I would suggest comparing not only confusion matrices, but also standard metrics for object detection (such as mAP) which take into account also correct localization and false positives.

   `Answer:` We agree that including standard object detection metrics provides a more complete picture of model performance. We have now included standard object detection metrics – specifically *mAP@0.5* – in Appendix C and have added a discussion about them in Section 7 ("Discussion"). *YOLOv5* models achieved high overall performance on the *Super-CLEVR* dataset, with mAP@0.5 values approaching 1.0 in all experiments. However, the goal of this work was not to maximise general object detection metrics, but rather to diagnose and correct specific systematic errors known as rare slices. For this purpose, per-class recall serves as a more precise diagnostic tool than a global metric like mAP. While a high mAP score confirms the good overall model performance, it can mask the poor performance on a specific, underrepresented slice of data, as the error is averaged out. By focusing on the recall of the problematic classes, we can directly measure the impact of the slice and, more importantly, verify the success of the mending process in a targeted manner. While the main analysis focuses on recall to clearly illustrate the diagnosis and repair of rare slices, a more comprehensive set of performance metrics is provided for completeness. We have included detailed results in Appendix C, which contains the confusion matrices, F1-Confidence curves, and other model training and validation performance metrics (e.g., mAP@0.5) for all *Super-CLEVR* hierarchies, both before and after model mending. This supplementary data confirms that the targeted improvements in recall are accompanied by corresponding positive gains in the F1-score, reinforcing the overall efficacy of the proposed SDM pipeline.

3 Review 2

1. It is mentioned that FastLAS allows for a penalty setting, but how is this penalty set? Is it done manually?

   `Answer:` The penalty values for positive and negative examples were set empirically, which is a standard practice for such hyperparameters. Our choice was guided by the need to address the significant class imbalance inherent in the rule learning task, where there are few positive examples (the misclassified rare slices) compared to a large number of negative examples. Consequently, we assigned a higher penalty to positive examples than to negative ones (e.g., penalties of 4 and 2, respectively, in our experiments). This forces *FastLAS* to prioritise finding rules that cover these few, but important, positive examples that characterise the rare slices. This approach ensures the learning process is focused on discovering meaningful and accurate logical rules for the rare slices. The rule head penalty in *FastLAS* was also set empirically to explore the effect of this hyperparameter on the quality of the extracted rules. These justifications have now been added to the "Rule Extraction and Selection" paragraph in Section 6.2.1 ("Experimental Setup").

2. Three ILP systems were used for rule extraction in the experimentation phase, each with their parameters, but their parameter selection is not justified.

`Answer:` In our study, the goal in this phase was not to find an optimal hyperparameter setting for each ILP system, but rather to ensure that our findings were robust and not an artefact of a specific, arbitrary configuration. All hyperparameter values were empirically fine-tuned by exploratory experimentation. Specifically, we selected several reasonable values to test different configurations of ILP systems in extracting meaningful rules describing rare slices. We acknowledge that manual parameter selection may not always be ideal, and future work could explore automated methods for setting hyperparameters. We have clarified this in the revised manuscript in the "Rule Extraction and Selection" paragraph in Section 6.2.1 ("Experimental Setup") and in Section 8 ("Conclusion and Future Work").

3. The model mending part I believe is one of the most valuable contributions of the paper, but there is not a lot of insight on this part. How is the mending done? Is it iteratively? What is the final impact?

   `Answer:` We have updated the model mending paragraphs in Section 6 ("Experiments") to provide a clear description of the process. Model mending is performed through a targeted data augmentation strategy. Based on the extracted rules by the ILP systems (e.g., "an image is difficult for the object detection model if it contains a utility bike facing north" in *Super-CLEVR*), we add new images to the training set that match these hypotheses. The defective model is then retrained on this augmented dataset. We tested several retraining epochs to find the most effective one for each case. The process of our SDM pipeline is iterative. As is now detailed in Sections 6.2.2 and 6.3.2 ("Experimental Results"), we demonstrate this by performing two iterations of the SDM on both the *Super-CLEVR* and *ImageNet* experiments. The second iteration shows how our framework can be applied again to diagnose and repair the more subtle, persistent errors that may remain after an initial model mending. The final impact was a significant and consistent improvement in model performance on the previously problematic classes. We have expanded Sections 6.2.2 and 6.3.2 ("Experimental Results") and Section 7.2 ("Impact of Model Mending") to better quantify this impact. To provide some illustrative examples:

   - In the *Super-CLEVR* (object detection) experiment, the recall for the "urban bicycle" class improved from 80.00% to 94.00% after the first model mending iteration, and further to 98.00% after the second.
   - In the *ImageNet* (image classification) experiment, the first model mending iteration significantly improved the Top-1 accuracy of the four problematic classes, raising them from as low as 62.25% to over 90.00%.

   These results, demonstrated across two different datasets and CV tasks, confirm that our model mending process leads to substantial and targeted performance gains.

4. Finally, it would have been interesting to see a comparison of the performance of the architecture across different datasets, since only one is considered and it can give a biased perception on the behavior of the model.

   `Answer:` To address this, in Section 6.3 ("*ImageNet* Experiments"), we have added new experiments on the real-world image dataset *ImageNet* to validate our SDM on an image classification task. This complements our original experiments on the synthetic *Super-CLEVR*

dataset, which focused on object detection, and demonstrates the versatility of the SDM approach across different Computer Vision (CV) tasks and its effectiveness in a more challenging and realistic domain.

4 Review 3

4.1 Problem Definition

1. I imagine that in [16] the definition of a slice contains the sentence "the model performs poorly" but this is ambiguous, and a criterion should be defined/adopted. This can be seen in Fig. 9 and Section 6.2.1, where no reason is given why those particular classes of objects are rare slices. Which is the chosen criterion? Last 5 performant classes? Classes with a recall lower than a threshold? I think that this part should be clearer with a better definition/criterion for rare slices.

   `Answer:` A rare slice is a subset of data, sharing a set of attributes, on which the model underperforms. It is induced by providing specific vehicle subclasses with a low occurrence probability in the training set. To make the "underperforms" condition concrete and measurable, we have now clarified that we adopt a formal criterion. A low-frequency vehicle subclass is considered a problematic rare slice if model performance on its parent target class falls at or below a dataset-dependent target class threshold τ_c. As we now motivate in the text, this threshold is determined for each experiment (e.g., 95.00% recall for *Super-CLEVR* and 86.00% Top-1 accuracy for *ImageNet*). This definition provides a clear and systematic method for identifying underperforming classes that possibly contain rare slices, making our methodology transparent and reproducible. We have clarified this point in the revised manuscript in Section 5 ("Rare Slice Generation Methodology").

2. Regarding Fig. 9 again, I would have expected lower performance for the rare slices. I would not call a class with 80% of recall as rare slice.

   `Answer:` We agree that an 80.00% recall value might seem acceptable in many standard, complex benchmarks. However, the significance of this score is relative to the specific context and high performance baseline of our experiments. As now clarified in Section 5 ("Rare Slice Generation Methodology") and in the "Rare Slice Generation and Initial Model Training" paragraph of Section 6.2.2 ("Experimental Results"), our methodology operates on the premise that the *YOLOv5* model is expected to achieve near-perfect ($> 95.00\%$) recall on all *Super-CLEVR* target classes under normal conditions, a baseline that is consistently met by the non-problematic classes in our experiments. Therefore, we define "underperformance" not in absolute terms, but as a significant deviation from this high baseline. We formalize this using the target class threshold τ_c, which is set at 95.00% for the *Super-CLEVR* experiments. A drop to 80.00% recall is not a minor fluctuation but a clear underperforming class to be investigated with our SDM.

3. In hierarchies 1 and 2 of the VT taxonomy there are no rare slices "as expected". But why did you expect this?

   `Answer:` Our taxonomy-based heuristic hinges on the fact that problematic rare slices are not just a result of low occurrence probability, but are most effectively induced when a model is

forced to distinguish between visually similar subclasses that have been placed into different target classes. To validate this heuristic, we designed the hierarchies in the following way:

- In *VT:$\mathcal{H}$3* and *VT:$\mathcal{H}$4*, we applied the heuristic to separate visually similar vehicle pairs (e.g., "dirtbike" and "mountain bike") into different target classes ("motorcycle" and "bicycle", respectively). As predicted by our heuristic, these hierarchies successfully induced problematic rare slices.
- In *VT:$\mathcal{H}$1* and *VT:$\mathcal{H}$2*, we deliberately did not apply the heuristic. In these hierarchies, visually similar vehicle pairs are always grouped together within the same parent class (e.g., both "dirtbike" and "mountain bike" fall under the general "land vehicle" class).

Therefore, we expected no problematic rare slices to emerge in *VT:$\mathcal{H}$1* and *VT:$\mathcal{H}$2*. Even though the training set did contain vehicle subclasses with low occurrence probability, the model trained on these hierarchies did not underperform. This demonstrates that the taxonomy structure is critical in creating scenarios that challenge a model with problematic rare slices, as low occurrence probability alone is insufficient to cause a failure. We have updated the "Taxonomies" paragraph in Section 6.2.1 ("Experimental Setup") and the "Rare Slice Generation and Initial Model Training" paragraph in Section 6.2.2 ("Experimental Results").

4.2 Paper Contextualization

1. The related work focuses on slice discovery methods and ILP. Where the former is the focus of the paper, the latter is just a background that (in my opinion) should be reduced and moved in the background section.

   `Answer:` We have now revised Section 2 ("Related Work") to focus on slice discovery methods and have moved and reduced the discussion of ILP to Section 3 ("Preliminaries").

2. In the related work about slice discovery methods, a comparison with other works showing how the proposal addresses open problems not addressed in precedence would help the reader in a better contextualization of the paper.

   `Answer:` We have now revised Section 2 ("Related Work") and Section 7.3 ("Comparison with Existing Methods") to emphasise this point. Recent methods in slice discovery and rare data mining, such as *Domino* and Talisman, have introduced strategies to identify rare or underperforming data regions by operating largely in embedding spaces or latent distributions. While effective, these approaches lack interpretability. For example, Jiang, Chiyu Max et al. (2022) proposed density-based *rare example mining* using normalizing flows over learned detection features in a 3D object detection setting. Although this approach significantly improves performance on rare intraclass instances, it does not provide semantic explanations of errors or provides insight into the nature of failure modes. In contrast, our neurosymbolic framework extracts interpretable logical rules that characterise performance drops on semantically coherent slices. These symbolic rules not only support direct error attribution and human-in-the-loop model debugging, but also enable targeted data augmentation and model mending. Compared to baselines in active learning or distributional mining, our SDM thus offers the advantage of transparency and editability, allowing model correction based on explicit domain-level logical rules.

3. The paper would be improved if contextualized with respect to the kind of approaches used, that is Neurosymbolic AI and mining of discriminative knowledge from pos/neg example. Regarding Neurosymbolic AI, this approach is quite different from approaches that embed the logic in neural networks or embed some differentiable function in logic systems. It would be interesting a discussion on what kind of NeSy integration this paper proposes. To this extent, the Kautz's taxonomy could be helpful, see Section 2 of the paper at `https://arxiv.org/pdf/2105.05330`. Regarding mining of discriminative knowledge from pos/neg examples, a recent paper does something similar for characterizing pos/neg examples of temporal traces, see the paper titled "Making Sense of Temporal Event Data: A Framework for Comparing Techniques for the Discovery of Discriminative Temporal Patterns" (Di Francescomarino et al., CAiSE 2024). It would be interesting to contextualize the present paper with respect to this trend of research.

   `Answer:` We have integrated this suggestion in Section 4 ("Neurosymbolic Framework for Slice Discovery"). According to Kautz's taxonomy of neurosymbolic systems, our SDM approach aligns with the [*Neuro→Symbolic*] paradigm, where the outputs of a neural system (here, a CV model) are post-processed by a symbolic module to derive explainable rules. Also, we thank the reviewer for the pointer to the recent work by Di Francescomarino et al. (CAiSE 2024). In Section 2 ("Related Work"), we have now added a discussion that contextualizes our work within this line of research. We now cite several related works that use logic-based methods in different domains, such as temporal data. We clarify that our method follows this trend but specifically focuses on using ILP for discovering explainable rules in the context of slice discovery for high-dimensional visual data. This helps to connect our specific application to a broader, active research area in explainable AI.

4.3 Results

1. The results are measured only according to the recall (if I understand correctly as it is not specified in the confusion matrixes) but there is no argumentation why only recall has been chosen. I would appreciate to see also the precision and F1 results as, after model mending, to a higher recall could correspond a lower precision.

   `Answer:` As stated in a previous comment, we agree that including standard object detection metrics provides a more complete picture of model performance. We have now included standard object detection metrics – specifically *mAP@0.5* – in Appendix C and have added a discussion about them in Section 7 ("Discussion"). *YOLOv5* models achieved high overall performance on the *Super-CLEVR* dataset, with mAP@0.5 values approaching 1.0 in all experiments. However, the goal of this work was not to maximise general object detection metrics, but rather to diagnose and correct specific systematic errors known as rare slices. For this purpose, per-class recall serves as a more precise diagnostic tool than a global metric like mAP. While a high mAP score confirms the good overall model performance, it can mask the poor performance on a specific, underrepresented slice of data, as the error is averaged out. By focusing on the recall of the problematic classes, we can directly measure the impact of the slice and, more importantly, verify the success of the mending process in a targeted manner. While the main analysis focuses on recall to clearly illustrate the diagnosis and repair of rare slices, a more comprehensive set of performance metrics is provided for completeness. We have included detailed results in Appendix C, which contains the confusion matrices, F1-Confidence

curves, and other model training and validation performance metrics (e.g., mAP@0.5) for all *Super-CLEVR* hierarchies, both before and after model mending. This supplementary data confirms that the targeted improvements in recall are accompanied by corresponding positive gains in the F1-score, reinforcing the overall efficacy of the proposed SDM pipeline.

2. I would discuss more the impact of the rule head penalty as, for some classes, certain values bring to wrong or no rules. The exception ration, instead, does not seem to impact the discovery. Please specify this.

   `Answer:` We have now added a discussion in Section 7.1 ("Comparison of ILP Systems") to address this point and clarify the sensitivity of ILP hyperparameters. As it was correctly noted, experimental results show that the rule head penalty in *FastLAS* has a significant impact on its ability to discover rare slices. We now discuss that lower penalty values consistently produced meaningful rules, whereas higher values sometimes prevented the system from discovering any rules. Conversely, we confirm the observation that the exception ratio in *FOLD-R++* had a minimal impact on its rule extraction, indicating limited sensitivity to this hyperparameter in our context.

3. The paper shows the confusion matrix only for the recall of the VT hierarchy 4. Other results (precision, recall, F1 for all the hierarchies VT 3, VT 4 and PP 1 before and after model mending) would make the paper more self-contained if included as appendixes.

   `Answer:` A more comprehensive set of performance metrics is now provided for completeness in Appendix C, which contains the confusion matrices, F1-Confidence curves, and other model training and validation performance metrics (e.g., mAP@0.5) for all *Super-CLEVR* hierarchies, both before and after model mending.

4.4 Limitations

1. I find the method highly tailored to the Super-Clever image generator and to the scene graph generation tasks. There is no discussion how the extracted rules can be used for other methods of synthetic-image generation for generating images of a different domain. In addition, it seems to me that the method is applicable to only images where a scene graph can be extracted from. Therefore, it can be hard to extend the method to, for example, medical images coming from radiographies. This kind of images cannot always be traduced in a scene graph. If this is the case, the example of chest X-rays in the introduction can be misleading and should be changed.

   `Answer:` Image generators are typically available for synthetic datasets and can easily adopt our methodology, as it only requires adjusting the occurrence probability of specific objects. In Section 6 ("Experiments"), we have now included a new set of experiments on the real-world *ImageNet* dataset, which has neither a synthetic image generator nor ground-truth scene graphs. For the *ImageNet* experiments, we simulated the data generation process by carefully subsampling from the complete dataset to construct splits with controlled distributions of rare slices. Notably, we generated the scene graph descriptions of the images using *GPT-4.1* as a Vision Language Model (VLM). The new experiments demonstrate that our SDM pipeline is not tied to a specific data generator. The logical rules extracted by the ILP systems can guide any data augmentation process, whether it is through a synthetic generator, controlled subsampling, or generative models. We believe that scene graphs can, in principle,

be generated for any image domain, although we acknowledge that this may be more difficult in areas such as medical imaging. Therefore, we have followed the suggestion and replaced the chest X-rays example from Section 1 ("Introduction") with a more appropriate one.

2. In addition, a section showing the limitations of the approach can help the reader to understand what the method does not address (or it does with difficulties) and potentially can foster further research.

   `Answer:` Following this advice, we have added Section 7.4 ("Limitations") that discusses the main limitations of the proposed SDM approach. We acknowledge that our SDM relies on the availability of scene graph representations to extract the logical rules identifying rare slices. We clarify that while scene graphs are not usually available for real-world datasets, this limitation is becoming increasingly tractable with the recent successful development of VLMs to fully automate this step, a key direction for our future work. A second limitation is the current need for manual, exploratory tuning for the hyperparameters of the ILP systems, which can be time-consuming and may require domain expertise. Furthermore, the scalability of ILP systems can be computationally intensive, especially with large validation sets or with a complex hypothesis space defined by numerous attributes and predicates, as observed with some timeouts in our experiments. Finally, the current implementation of our SDM focuses on discovering rare slices defined by object attributes (e.g., "a yellow rubber utility bike facing south"). Extending it to include rare slices defined by the relationships between objects (e.g., "a bicycle next to a car") remains a key direction for future work.

4.5 Presentation

1. The structure of the paper is good but sometimes I feel lost without a proper running example. There are some examples but not always connected among them. I strongly suggest using a running example.

   `Answer:` To address this, we have revised the manuscript to incorporate a running example throughout the paper. This example, drawn from our *Super-CLEVR* experiments, follows a specific underperforming rare slice: a "utility bike" with certain attributes (e.g., facing south) that the model systematically misclassifies. This example now appears in Sections 3.1 ("Super-CLEVR"), 3.3 ("Inductive Logic Programming"), 4.4 ("Rule Extraction via Inductive Logic Programming"), and 4.5 ("Model Mending") illustrating the ILP encodings, rule extraction process, and model mending impact.

2. Section 3.2: What is the expressivity of the language of B, h and E? Fully propositional, First-Order? Please specify it for all the three methods.

   `Answer:` All three ILP systems are first-order and operate using sets of positive and negative ground examples. We specified this in Section 3.3 ("Inductive Logic Programming").

3. Page 6: two different symbols are used for bounding boxes: b and $\mathcal{B}$, please adjust.

   `Answer:` In Section 4.2 ("Object Detection and Image Classification"), we adjusted the symbols using $\hat{b}$ for the predicted bounding box, and $\hat{h}$ for the predicted class.

4. Section 4.1 the class dirtbike can have many "root classes" according to Fig. 6. Why has the class "motorcycle" been chosen? I am afraid I missed something.

`Answer:` Yes, the "dirtbike" subclass can indeed have multiple potential root classes depending on the specific hierarchy $\mathcal{H}$ being considered. The choice of "motorcycle" was an example of a possible root class for "dirtbike" to clarify the concept of h_i. As noted, and as shown in Fig. 7, different hierarchies selected from the *Vehicle Type* taxonomy could assign different root classes. To avoid confusion, we have clarified this in the revised sentence in Section 4.1 ("Data Generation"), explicitly mentioning that the root class depends on the hierarchy $\mathcal{H}$ under consideration and providing alternative possibilities like "land vehicle" alongside "motorcycle".

5. Page 7: How did you select positive and negative images? I guess there is a ground truth label for the whole image, but I cannot find its description. However, at the beginning of Section 4.2, an object detection problem is described, therefore I do not understand what a (un)correctly classified image is.

   `Answer:` Indeed, the task for the *Super-CLEVR* dataset is object detection, and the ground truth labels exist at the object level (bounding box and target class for each object), not as a single label for the whole image. The E_h^+ and E_h^- image sets are defined after running the model on the validation split and are specific to each root class h in a hierarchy H. They are based on the performance of a model f on the objects labelled h within an image:

   - An image is in E_h^+ (positive set for a class h) if the model misclassifies at least one object that has ground truth label h.
   - An image is in E_h^- (negative set for a class h) if the model correctly classifies all objects that have ground truth label h.

   Essentially, we aggregate the object-level classification results for a specific class h to categorise the entire image based on whether any misclassification occurred for that class within it. This categorisation is necessary for the subsequent step using ILP, which leverages positive and negative examples to identify rare slices of classes where the model underperforms. We have clarified this in the revised manuscript in Section 4.2 ("Object Detection and Image Classification").

6. Section 4.4: not clear to me the difference between GE+ and E+ILP. Why are they assembled?

   `Answer:`

   - $G_{E_h^+}$ resp. $G_{E_h^-}$: These represent the sets of scene graphs for the positive and negative examples, respectively, of class h. They are the graph-based representations of the images.
   - $ILP_{E_h^+}$ resp. $ILP_{E_h^-}$: These represent the sets of positive and negative examples translated into their logical representation suitable for an ILP system. This involves converting the objects and their attributes depicted in the scene graphs into logical facts.

   We have revised and improved the explanation in Section 4.4 ("Rule Extraction via Inductive Logic Programming") to clarify this.

7. In Section 4.4 I get lost when Figure 3 is described. Here I feel the need of a running example for a better understanding of the pos/neg examples. In general, the background knowledge (BK) should state general common sense information, such as, if A is next to B then B is

next to A, but here there is a not defined "contains(19, 0)". The mode bias seems a more suitable candidate for BK but, unfortunately, there is no definition of what a mode bias for non-experts in ILP is.

`Answer:` We have revised the manuscript to incorporate a running example throughout the paper. This example, drawn from our *Super-CLEVR* experiments, follows a specific underperforming rare slice: a "utility bike" with certain attributes (e.g., facing south) that the model systematically misclassifies. This example now appears in Sections 3.1 ("Super-CLEVR"), 3.3 ("Inductive Logic Programming"), 4.4 ("Rule Extraction via Inductive Logic Programming"), and 4.5 ("Model Mending") illustrating the ILP encodings, rule extraction process, and model mending impact. Then, we have revised Section 4.4 ("Rule Extraction via Inductive Logic Programming") to clarify that our SDM approach uses *context-dependent* background knowledge, particularly with *FastLAS*. This means the background knowledge is not a set of general rules (like `next_to(A,B) -> next_to(B,A)`), but rather a set of scene-specific facts that describe a single image (e.g., `contains(19, 0)` means "scene 19 contains object 0", and `shape(0, utility)` means "object 0 is a utility bike"). This context-dependent background knowledge is directly derived from the scene graph of each image and is provided to the ILP system alongside each positive or negative example. We have also added an explanation of mode bias. The mode bias defines the structure of hypotheses that the ILP system can consider. In essence, it acts as a guide for the hypothesis search, telling the ILP system what predicates can appear in the head (`#modeh`, in *FastLAS*) versus the body (`#modeb`, in *FastLAS*) of a rule, how many variables can be used in a rule, and so on. This helps prune the search space and ensures that the learned rules are semantically meaningful for our task.

8. Page 9: What do you mean with "... applying it, with appropriate adjustments, in similar applications settings is suggestive"? Please use a more precise and formal wording as required in a scientific paper.

   `Answer:` We have revised this sentence in Section 5 ("Rare Slice Generation Methodology") to make it more precise and better reflect our intended meaning regarding the potential of the methodology for other application domains.

9. Page 9, bullet 4: How are other attributes (e.g., material, shape, color) chosen? Randomly?

   `Answer:` For rare slices, the user may restrict each subclass in S by specifying any combination of attribute values that makes the respective slice more specific. For example, a rare slice can be defined as the "dirtbike" subclass with colour "red" and material "metal". These user-specified attributes are exhaustively combined with all values of the remaining attributes. For example, if the attribute "size" is not specified, then the rare slice "dirtbike-red-metal" will include all possible values of "size", i.e. "dirtbike-red-metal-small" and "dirtbike-red-metal-large". Non-rare slices consist of all remaining combinations of subclasses and attribute values that are not rare slices. For example, the combination "dirtbike-blue-metal-small" is a non-rare slice because the user has restricted the rare slice to the colour "red". We have clarified this point in the revised manuscript in Section 5 ("Rare Slice Generation Methodology").

10. Caption of Table 1: why did you test only on the models trained for 160 epochs and not on the models trained for 80 and 320 epochs? It seems that these models are never used. Why did you use the plural in neural network models? I thought you trained only one Yolov5 model for all the classes in the hierarchies. If this is not the case, please explicit state it. What is

the criterion for wrong rules that you used for the X symbol?

`Answer:` We did not train only one *YOLOv5* model. Instead, we trained a distinct *YOLOv5* model for each of the five hierarchies (VT:$\mathcal{H}1$ – VT:$\mathcal{H}4$ and PP:$\mathcal{H}1$), as each hierarchy represents a unique classification task with a different set of target classes. The problematic rare slices appeared consistently across all training epochs, i.e. 80, 160, and 320. The models trained for 160 epochs were selected for slice discovery because they yielded the best initial classification results on the validation set across the different hierarchies. We have updated Sections 6.2.1 and 6.2.2 ("Experimental Setup" and "Experimental Results") to be explicit on these points. Furthermore, in the "First Rule Extraction and Selection Iteration" paragraph within Section 6.2.2 ("Experimental Results"), we have updated the text introducing the rule extraction tables to clarify the meaning of their symbols. As explained, an entry is marked with a ✓ if at least one rule extracted by the ILP system agrees with a candidate rule for the target class, while a ✗ indicates that, under a specific configuration, no extracted rule does. As detailed in the "Rule Extraction and Selection" paragraph in Section 6.2.1 ("Experimental Setup"), to formalise which of the candidate rules to consider as descriptions of potential rare slices, we introduce the rare slice hypothesis threshold τ_h. Consequently, only candidate rules that agree with a percentage of extracted rules greater than or equal to τ_h are retained.

11. Page 14, second to last line: Popper fails with offroad car, offroad vehicles and specialized vehicles (Table 1) and not with pickup truck and articulated bus as in the sentence.

    `Answer:` This explanation is now clearer following a significant revision of the entire Section 6 ("Experiments") to improve its clarity, structure, and exposition. We believe the updated section is now more rigorous, easier to follow, and transparently details our methodology and results.

12. What is a "native scene graph" mentioned at page 16? Please clarify this.

    `Answer:` The term "native scene graph" was intended to refer to scene graphs that are generated directly from raw images, as opposed to the ground-truth scene graphs that are synthetically produced alongside the images by the *Super-CLEVR* generator. To avoid this ambiguity, we have removed such a term and rewritten the paragraph in Section 8 ("Conclusion and Future Work") more explicitly. The revised text now clarifies that a key direction for future work is to systematically explore and integrate VLMs to our SDM to fully automate the generation of scene graphs for slice discovery.

4.6 Other (not minor) technicalities

1. Two concerns regard the model mending. In section 6.1, in the model mending paragraph only 12 new images are generated. For me 12 new images for slice are totally irrelevant in a training set of 10K images. If 12 images are sufficient, this is a result that deserves more discussion. The second concern regards the extracted rules. My understanding of Fig. 5 is that hard(V0) is described by the first rule OR the second OR the third. Is this the case? If so, how did you encode this OR in the image generator? In general, there is no detailed description about how the mined rules are translated into specifications for the image generator.

   `Answer:` In our revised manuscript, we have improved the experimental setting regarding the model mending process. The goal was to solve the data imbalance by augmenting the original training set with new images based on the selected rules, so as to avoid catastrophic forgetting

in the CV model. Based on this, we determined that a more substantial number of new images was required. For the *Super-CLEVR* experiments, we augmented the training set with 500 new images for each identified rare slice. For the *ImageNet* experiments, we followed the same model mending process, augmenting the dataset with a similar number of new images for each slice to balance the subclass distribution. Your understanding of Fig. 4 (previously Fig. 5) is correct: the extracted rules for a rare slice represent a logical *OR* (e.g., "a utility bike facing north" OR "a utility bike facing south"). Note that the dataset augmentation is done according to all the candidate rules that agree with a percentage of extracted rules greater than or equal to the rare slice hypothesis threshold τ_h. We have now added a more detailed description of how these mined rules are translated into specifications in the model mending paragraphs in Section 6 (Experiments).

2. Fig. 2 shows the system architecture that is interesting as it is a closed loop. Therefore, it would be interesting having experiments with more cycles of this loop (at least 2) and see whether performance increase.

   `Answer:` To address this, in the revised manuscript in Section 6 ("Experiments") and Appendices C and D, we have significantly extended our experimental evaluation to include a second iteration of the SDM pipeline for both the *Super-CLEVR* and *ImageNet* experiments. Experimental results show that this pipeline is effective when applied iteratively. After the first model mending iteration, we evaluated again the models, identified the new underperforming classes, and applied the rule extraction and model mending process a second time. Experimental results show that this second iteration led to further performance improvements. For example, in $VT\mathcal{H}4$, the recall on the "urban bicycle" class increased from 94.00% to 98.00%, successfully resolving the persistent rare slice regarding the "utility bike" vehicle subclass. This shows the effectiveness of our closed-loop pipeline for slice discovery and model mending.