

# Active Learning in Neurosymbolic AI with Embed2Sym

Alexander Philipp Rader<sup>1</sup>, Alessandra Russo<sup>1</sup>

<sup>1</sup>Imperial College London

## Abstract

Neurosymbolic AI combines neural networks with symbolic reasoners in an effort to create robust and logical machine learning frameworks. In one approach, a neural component processes raw data and outputs latent concepts. A symbolic component then conducts logical reasoning with the concepts to produce the final result. A major hurdle lies in the propagation of the end label signal to the latent space when no latent labels are available. We investigate the use of active learning to alleviate this problem. In particular, we consider the neurosymbolic framework Embed2Sym. We adapt the learning framework to incorporate active learning by gaining a latent learning signal for misclassified examples. An oracle, such as a human in the loop, provides latent labels, which are used to finetune the neural component. Using the same benchmark datasets as the original paper, we empirically evaluate our method. We demonstrate that even a small amount of labelled latent data leads to a sizeable increase in accuracy.

## Keywords

Neurosymbolic AI, active learning, human-in-the-loop

## 1. Introduction

Neurosymbolic AI aims to combine the robustness of neural networks to real-world data with the explainability and provable correctness of symbolic reasoners [1]. A particular stream is known as "[Neuro  $\rightarrow$  Symbolic]" [2] and is reminiscent of the two-system model of the human mind [3]. The neural network represents system 1 and processes raw inputs to produce latent concepts. The symbolic component then logically reasons over the concepts, such as in system 2, to solve the given problem.

One such framework is Embed2Sym [4]. It consists of a neural network that transforms raw inputs into embeddings, a clustering algorithm that assigns them categories, and a symbolic optimiser based on answer set programming (ASP), that solves a logical task. One of the biggest challenges for Embed2Sym, and [Neuro  $\rightarrow$  Symbolic] architectures in general, is training the neural component without labels for the intermediate representations.

In this paper, we propose to mitigate this problem by providing latent signals using active learning, which allows the system to ask an oracle to annotate datapoints [5]. We extend Embed2Sym to incorporate active learning for incorrectly classified examples. We investigate the effect in three tasks: MNIST addition, CIFAR10 addition, and Member. Despite providing only a small percentage of latent labels, we attain substantial accuracy improvements.


---

*Cognitive AI 2023, 13th-15th November, 2023, Bari, Italy.*

✉ [apr20@ic.ac.uk](mailto:apr20@ic.ac.uk) (A. P. Rader); [a.russo@ic.ac.uk](mailto:a.russo@ic.ac.uk) (A. Russo)



© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

## 2. Background

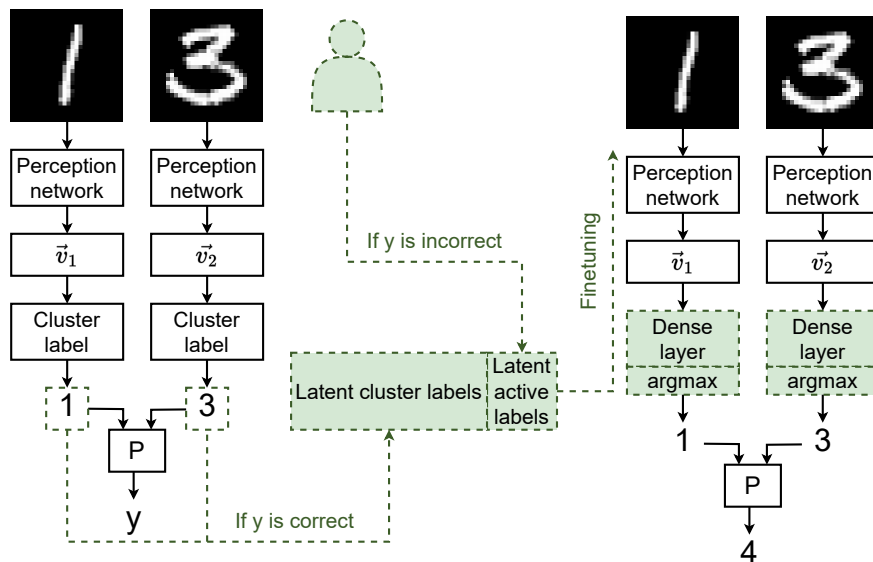
### 2.1. Task formulation

Each task contains raw and symbolic inputs and labels representing the result of a downstream operation. For example, in MNIST Addition the input consists of images of two numbers and the output equates to their sum. Crucially, no labels are provided for the intermediate, called *latent*, representations. In our example, there are no labels for the numbers themselves, only their sum.

### 2.2. Embed2Sym

The Embed2Sym framework contains a neural perception component and a symbolic reasoning component. The system works in three stages: [4]

1. **Fully neural model:** A neural network is trained end-to-end on the downstream task. It is a two-stage architecture containing a perception and a reasoning component, both of which are neural. Each input is processed by the perception network which creates embedding vectors. The reasoning network processes the concatenated embeddings to output the predicted label.
2. **Cluster discovery** The k-means algorithm divides the embedding space created by the perception network into clusters. The number of clusters is predetermined.
3. **Cluster labelling** An ASP algorithm assigns each cluster their symbolic meaning by means of an optimisation task. It utilises a hard-coded symbolic component to compute the downstream result from the latent concepts.



**Figure 1:** Embed2Sym framework on the left, active extension on the right, where additions are shown as green dotted lines

At inference time, the algorithm works in three steps, as illustrated on the left side in Figure 1. First, the neural perception network turns the inputs into embeddings  $\vec{v}_i$ . Second, the clustering assigns each embedding a symbolic label. Third, the hard-coded symbolic reasoning component  $P$  calculates the end result.

### 3. Active Embed2Sym

The neural component in Embed2Sym generates embeddings of the latent concepts, for which it has no labels. Instead, it uses downstream labels for training, which is a more difficult task. The core idea of this paper is to finetune the neural network with latent labels after it has been trained end-to-end. We acquire the latent labels from two sources, as shown in Figure 1:

1. For all examples with a correct end prediction, we assume that the predicted latent concepts are correct as well. Therefore, we can use the cluster labels from the trained perception networks. This represents the vast majority of examples.
2. For all examples with an incorrect end prediction, we ask an oracle for the corresponding latent labels. We refer to these as *active labels*. For complex tasks, the oracle is typically a human in the loop. In our case, we can use existing labels for MNIST and CIFAR10 images.

Since the last layer in the original framework uses k-means clustering, it is not differentiable. Therefore, we replace the clustering layer with a multi-layer-perceptron, indicated by the green boxes labelled "dense layer".

## 4. Results

We assess the effect of our extension by performing the tasks outlined in the original paper: MNIST and CIFAR10 addition, as well as Member. In the first two tasks, the inputs consist of images representing numbers and the output indicates their sum. In the member task, the input consists of an MNIST image and a list of numbers. The output is a binary variable indicating whether the number is a member of the list. [6]

We investigate two main questions:

1. What proportion of labels needs to be obtained by an oracle?
2. Does active learning improve the accuracy of the results?

We conducted each experiment for five independent runs and show the average scores and their standard deviations.

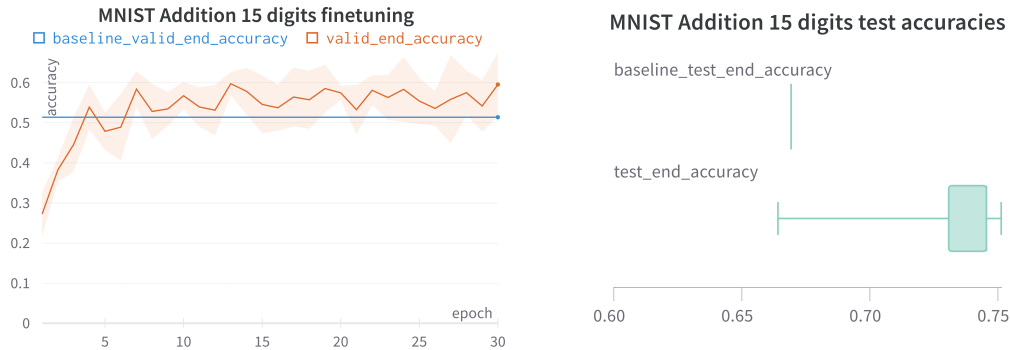
### 4.1. Proportion of active labels

Even though the latent labels for these specific tasks are easy to come by, calls to oracles are generally very expensive. Therefore, we need to evaluate our results in light of the percentage of active labels utilised.

Table 1 presents the number of active labels used for each task, as well as their proportion of the dataset. In each case, the percentages remain below 5%. We conclude that the number

**Table 1**  
Active labels

Task	Number of active labels	Percentage of dataset
MNIST Addition 1 digit	526	0.9%
MNIST Addition 15 digits	2723	4.6%
CIFAR10 Addition 1 digit	1717	3.6%
Member 3 digits	269	0.9%
Member 20 digits	425	1.4%



**Figure 2:** MNIST Addition learning curves during finetuning

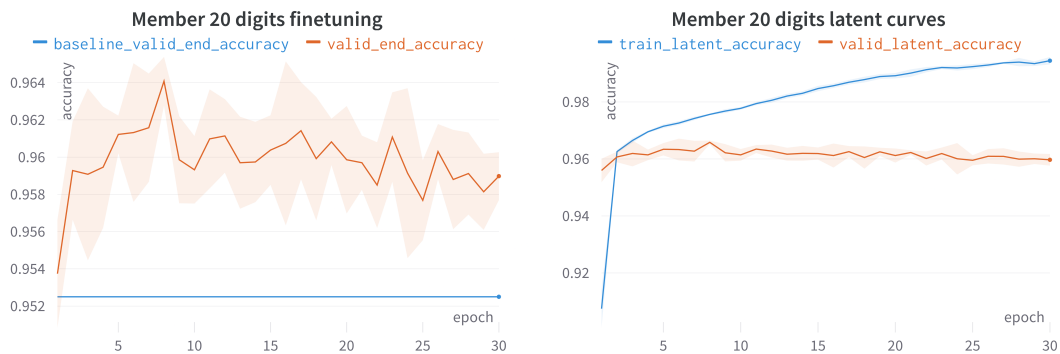
of necessary labels are feasible to obtain and the added effort is justifiable to achieve better accuracies.

## 4.2. Experimental results

Figure 2 shows the learning curves while finetuning the neural component for MNIST addition with 15 digits. The baseline is the accuracy achieved from the fully trained original framework. The extended model surpasses the baseline within a few number of epochs. The test accuracies are on average significantly higher than the baseline, with only one run being slightly below. Similar results occur for MNIST additions with a lower number of digits, as well as CIFAR10 addition, albeit not as pronounced.

In Member, the accuracy already surpasses the baseline after only round of training (epoch 0), as Figure 3 illustrates on the left. Interestingly, the accuracy starts to decline again after epoch 8. The graph on the right indicates that the model is overfitting, as the train and validation curves diverge. Further investigation reveals that 5.9% of incorrect latent labels were missed during the active labelling, because the downstream label was correct for them.

The nature of the member task facilitates correct end predictions despite wrong latent predictions in two major ways: First, the label is binary, so a random guess can achieve an accuracy of 50%. Second, most digits in each input list are irrelevant for the task. For example, let the list be [0,8,5] and the digit be 2. It does not matter whether any numbers in the list are misclassified, unless as a 2, the answer of "No" is still correct.



**Figure 3:** Member learning curves during finetuning

**Table 2**

Test set accuracies on the downstream tasks

Task	Embed2Sym	Active Embed2Sym
MNIST Addition 1 digit	0.97	<b>0.98±0.002</b>
MNIST Addition 15 digits	0.67	<b>0.73±0.036</b>
CIFAR10 Addition 1 digit	0.83	<b>0.88±0.004</b>
Member 3 digits	0.96	<b>0.98±0.003</b>
Member 20 digits	0.96	<b>0.97±0.002</b>

Table 2 summarises the results. There are 3 main takeaways from this investigation:

1. Active learning improved the accuracy in every task while requiring only a small percentage of oracle-annotated labels.
2. Active learning is most effective when there is more room for improvement. The greatest accuracy increase occurred on the dataset with the lowest baseline, MNIST 15.
3. False positives are an issue when the downstream labels are forgiving to mistakes in the latent space. This was the case in the Member tasks, where incorrect digit classifications often did not affect the outcome.

## 5. Conclusion

Active learning shows some promising results in our experiments. Providing a stronger signal in the latent space helped achieve a better performance, especially when the baseline had room for improvement. This work represents a first step towards investigating active learning in neurosymbolic AI. Future work includes extending other frameworks, such as [7], and solving more complex tasks. Using tasks that require human labelling will be able to demonstrate the wider impact of our proposal. Furthermore, we aim to use active learning also for symbolic rule learning. Embed2Sym hard-codes the rules, but for other frameworks we need to devise ways of providing active labels for them.

## Acknowledgments

This work was supported by UK Research and Innovation [grant number EP/S023356/1], in the UKRI Centre for Doctoral Training in Safe and Trusted Artificial Intelligence ([www.safeandtrustedai.org](http://www.safeandtrustedai.org)).

## References

- [1] A. Garcez, L. Lamb, Neurosymbolic ai: the 3rd wave, *Artificial Intelligence Review* (2023) 1–20. doi:10.1007/s10462-023-10448-w.
- [2] M. K. Sarker, L. Zhou, A. Eberhart, P. Hitzler, Neuro-symbolic artificial intelligence: Current trends, 2021. arXiv:2105.05330.
- [3] D. Kahneman, *Thinking, fast and slow*, Farrar, Straus and Giroux, New York, 2011. URL: [https://www.amazon.de/Thinking-Fast-Slow-Daniel-Kahneman/dp/0374275637/ref=wl\\_it\\_dp\\_o\\_pdT1\\_nS\\_nC?ie=UTF8&colid=151193SNGKJT9&coliid=I3OCESLZCVDFL7](https://www.amazon.de/Thinking-Fast-Slow-Daniel-Kahneman/dp/0374275637/ref=wl_it_dp_o_pdT1_nS_nC?ie=UTF8&colid=151193SNGKJT9&coliid=I3OCESLZCVDFL7).
- [4] Y. Aspis, K. Broda, J. Lobo, A. Russo, Embed2Sym - Scalable Neuro-Symbolic Reasoning via Clustered Embeddings, in: *Proceedings of the 19th International Conference on Principles of Knowledge Representation and Reasoning*, 2022, pp. 421–431. URL: <https://doi.org/10.24963/kr.2022/44>. doi:10.24963/kr.2022/44.
- [5] E. Mosqueira-Rey, E. Hernández-Pereira, D. Alonso-Ríos, J. Bobes-Bascarán, A. Fernández-Leal, Human-in-the-loop machine learning: a state of the art, *Artificial Intelligence Review* 56 (2022). doi:10.1007/s10462-022-10246-w.
- [6] E. Tsamoura, T. Hospedales, L. Michael, Neural-symbolic integration: A compositional perspective, *Proceedings of the AAAI Conference on Artificial Intelligence* 35 (2021) 5051–5060. URL: <https://ojs.aaai.org/index.php/AAAI/article/view/16639>. doi:10.1609/aaai.v35i6.16639.
- [7] D. Cunnington, M. Law, J. Lobo, A. Russo, Neuro-symbolic learning of answer set programs from raw data, 2023. arXiv:2205.12735.