Learning by Self-Explaining

Anonymous authors
Paper under double-blind review

Abstract

Artificial intelligence (AI) research has a long track record of drawing inspirations from findings from biology, in particular human intelligence. In contrast to current AI research that mainly treats explanations as a means for model inspection, a somewhat neglected finding from human psychology is the benefit of self-explaining in an agents' learning process. Motivated by this, we introduce a novel learning paradigm, termed **Learning by Self-Explaining** (LSX). The underlying idea is that a learning module (learner) performs a base task, e.g. image classification, and provides explanations to its decisions. An internal critic module next evaluates the quality of these explanations given the original task. Finally, the learner is refined with the critic's feedback and the loop is repeated as required. The intuition behind this is that an explanation is considered "good" if the critic can perform the same task given the respective explanation. Despite many implementation possibilities the structure of any LSX instantiation can be taxonomized based on four learning modules which we identify as: Fit, Explain, Reflect and Revise. In our work, we provide distinct instantiations of LSX for two different learner models, each illustrating different choices for the various LSX components. We broadly evaluate these on several datasets and show that Learning by Self-Explaining not only boosts the generalization abilities of AI models, particularly in small-data regimes, but also aids in mitigating the influence of confounding factors, as well as leading to more task-specific and faithful model explanations. Overall, our results provide experimental evidence of the potential of self-explaining within the learning phase of an AI model.

1 Introduction

Self-reflection is considered an important building block of human intelligence and a crucial component in the learning process and knowledge development of humans (Gläser-Zikuda, 2012; Ellis et al., 2014). In fact, one aspect of self-reflection—self-explaining—has been identified in several psychological studies as greatly beneficial for the overall learning, problem-solving and comprehension abilities of human subjects (Chi, 2018; Chi et al., 1981; 1994; Chamberland & Mamede, 2015; Belobrovy, 2018; Larsen et al., 2013; Kwon & Jonassen, 2011; Bisra et al., 2018). Accordingly, self-explanations act as a means of making initially implicit knowledge explicit and thereby allow for iterative and critical *self-refinement*.

Indeed, recent works in AI research have picked up on the idea of self-refining, either directly inspired by findings from human studies (Madaan et al., 2023) or otherwise motivated *e.g.*, by the potential of pre-trained large language models (LLMs), *e.g.*, on the topics of self-debiasing (Schick et al., 2021) and self-instructing (Wang et al., 2023). Although these works are quite specific in their form of self-refinement (*cf.* Pan et al. (2023) for a recent survey) and far from the general idea of self-reflection from human psychology, they provide valuable first steps for more *reflective AI*. However, none of these focus on the value and potential of (self-)explanations as the basis and means of such reflective processes. On the other hand, research on interactive machine learning (Teso et al., 2023; Gao et al., 2022) such as explanatory interactive learning (XIL) (Teso & Kersting, 2019; Schramowski et al., 2020) has long identified the value of explanations as a means of communication between human users and AI models and particularly as a means for model refinement. However, hereby explanations are only leveraged for refinement through human guidance.

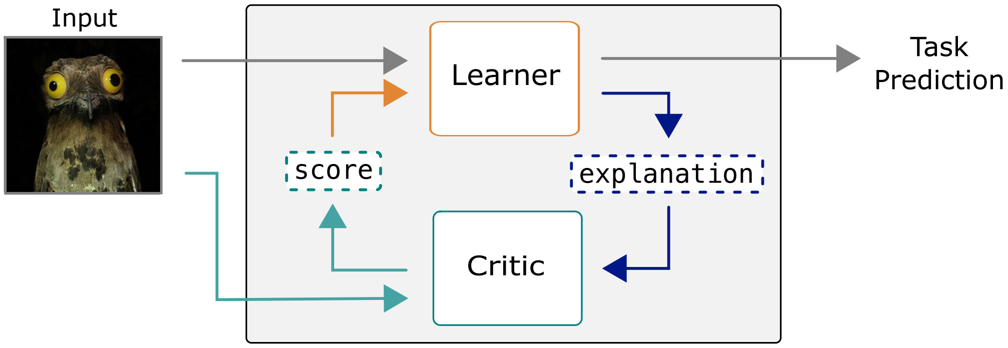

Figure 1: Learning by Self-Explaining (LSX) is a general learning framework that can be integrated into any base learning task, *e.g.*, image classification. It is characterized by two submodels, a *learner* and *critic*, and four distinct training modules: Fit, Explain, Reflect and Revise. Briefly, the learner is optimized for a base task in Fit, after which it provides `explanations` to its decisions in Explain. In the Reflect module these explanations are passed to the critic, which "reflects" on the quality of these explanations. In other words, the critic evaluates how useful the explanations are for performing the base task. The resulting feedback from the critic, `score`, is used to update the learner's representations in Revise. This loop can be repeated as needed.

In this work, we therefore introduce a novel machine learning paradigm called Learning by Self-Explaining (LSX) which leverages explanations in the learning phase of an AI model prior to any form of explanatory human guidance. The main idea is that an AI model consists of two submodels, a *learner* and a *critic*. The learning process in LSX is characterized by four learning modules sketched in Fig. 1. The learner is trained on a base task in Fit, after which it provides explanations for that task in Explain. The critic next performs the same task as the learner, but receives the input *and* corresponding explanations, thereby assessing the quality of these explanations for performing the original task. Intuitively, a "usefull" explanation should thereby provide important information for the task at hand. Finally, the critic's feedback is returned to the learner for revision in the Revise module and the Explain, Reflect and Revise loop is repeated, if needed. In the context of introducing LSX, we further present two instantiations of LSX for training a convolutional neural network (CNN) and the neuro-symbolic concept learner of Stammer et al. (2021) (NeSyCL), thereby illustrating specific configurations of the submodels and learning modules of LSX as well as the flexibility of the paradigm. In the context of our experimental evaluations on multiple datasets, we show that LSX boosts the generalisability of base learners, particularly in the small data regime. Moreover, we show that LSX helps mitigate the influence of confounding factors and leads to more consolidated, task-specific and faithful model explanations.

In summary, our contributions are the following: (i) We introduce a novel learning paradigm for machine learning, based on a model self-refining itself by evaluating its own explanations. (ii) We introduce two different instantiations of LSX, illustrating different submodel and learning module configuration choices. (iii) We provide extensive experimental evidence on various datasets and evaluation metrics, illustrating the benefits and potential of LSX.

We proceed as follows. In section 2, we formally introduce the LSX paradigm. In section 3, we next introduce two specific instantiations that integrate LSX into their training procedure. In our experimental evaluations in section 4, we provide results on various datasets and metrics illustrated via both LSX instantiations. We finish our work with an extensive discussion on related work and leave the reader with a final conclusion. [1]

2 Learning by Self-Explaining (LSX)

LSX is not explicitly bound to any one type of model implementation, data domain or base learning task, but can be considered a general learning approach for any base learner. In this section, we therefore give an overview of the basic set of modules that characterize LSX before continuing with two specific instantiations in the next section. Let us first provide the background notations.

---

[1] Code will be made available soon.

Algorithm 1 Learning by Self-Explaining: Given two submodels, a learner model (f) and internal critic model (c), input sample sets $\bar{X}_f = (X_f, y_f)$ and $\bar{X}_c = (X_c, y_c)$ (tuple sets which include raw input samples *e.g.*, X_f, and corresponding label sets *e.g.*, y_f), original task (e.g. image classification) and iteration budget T.

```
$f \leftarrow \textsc{Fit}(f, \bar{X}_f)$ {Learner optimized for base task}
repeat
    $E_c \leftarrow \textsc{Explain}(f, \bar{X}_c)$ {Obtain explanations from learner for examples $X_c$}
    $\texttt{score} \leftarrow \textsc{Reflect}(c, E_c, \bar{X}_c)$ {Critic provides feedback on the quality of the explanations}
    $f \leftarrow \textsc{Revise}(f, \bar{X}_f, \bar{X}_c, \texttt{score})$ {Learner is updated given feedback from critic}
until budget $T$ is exhausted or $f$ converged
return $f$
```

For simplicity, we introduce LSX here in the context of supervised image classification as base task. More formally, let $x_i \in X$ be an image, with the full dataset $X := [x_1, ..., x_N] \in \mathcal{R}^{N \times L \times M}$, and with corresponding class label to each x_i defined as $y_i \in \{1, ...K\}$. Hereby, N represents the number of data samples, L and M the image dimensions and K the maximal number of image classes. Furthermore, let X be split into a train and test split $X = \{(X_{train}, y_{train}), (X_{test}, y_{test})\}$ with y_{train} representing the set of corresponding image class labels of the training set and y_{test} those of test set. The learner is provided the tuple set $\bar{X}_f = (X_f, y_f) = (X_{train}, y_{train})$ where we denote $y_f = y_{train}$ specifically as the label set provided to the learner. The critic set, $\bar{X}_c$, can represent a subset of $\bar{X}_f$ *e.g.*, $\bar{X}_c \subseteq \bar{X}_f$, but also a previously separated set from the test set (details below). Also for the critic we specifically denote y_c as the label set of the $\bar{X}_c$.

The basic idea of LSX (*cf.* Fig. 1) is the following. An AI model consists of two submodels: a *learner*, f, and an internal *critic*, c. The learner can represent any desired learning model, e.g. a convolutional neural network (CNN). Its initial step is to perform a base task, *e.g.*, supervised image classification. Next, the critic reflects on the learner's explanations by evaluating the quality of these explanations for performing the initial task itself. If an explanation is "good", having this explanation up front will help solving the same task. Hereby, c can represent any AI model. After the critic has given feedback on the explanations, the learner revises its representations accordingly. This general procedure can be described via four modules Fit, Explain, Reflect and Revise, where the last three modules describe the core of LSX and can be repeated until an iteration budget T is reached. Let us now describe these four modules (presented in pseudo-code in Alg. 1) in more detail.

Base task (Fit). The Fit module describes the underlying, base learning task in which the learner is optimized to solve a particular problem, *e.g.*, supervised image classification. Overall, this module is independent of LSX as it represents the standard, data-driven machine learning approach. More specifically, in Fit, f is provided with the training dataset, $\bar{X}_f$, makes predictions given the base task, $\hat{y}_f = f(X_f)$, and optimizes its latent representations given the corresponding loss function, l_{B}. This loss function can *e.g.*, correspond to the cross-entropy loss, $l_{\text{B}} = l_{\text{CE}}(\hat{y}_f, y_f)$ when considering supervised classification. However, the underlying task can correspond to any form of learning (*e.g.*, other forms of supervision) and can contain any bells and whistles of modern machine learning setups (*e.g.*, hyperparameter optimization, learning rate schedulers, etc.). Finally, Fit returns a model optimized for the base task, $f = \textsc{Fit}(f, \bar{X}_f)$.

Obtain explanations (Explain). Explain represents the first of three core modules of LSX. In this module f provides explanations to its decisions given a set of data samples, X_c. This is achieved via a predefined explanation method which returns an `explanation` for each sample, $E_c = \textsc{Explain}(f, X_c)$, where $E_c := \{e_1, ..., e_{|X_c|}\}$. If $\bar{X}_c$ contains ground-truth annotations, explanations can be queried given the ground-truth labels ($E_c = \textsc{Explain}(f, X_c|y_c)$), otherwise the learner provides `explanations` given its predictions. Hereby $\bar{X}_c$ can be a subset of $\bar{X}_f$ (*i.e.*, $\bar{X}_c \subseteq \bar{X}_f$), but can also represent a separate set that is with-held during the initial optimization of f (*i.e.*, $\bar{X}_c \cap \bar{X}_f = \emptyset$). Overall, this explanation module can be realized with any explanation method from the vast literature of XAI (*cf.* Guidotti et al. (2018); Ras et al. (2022b); Liao & Varshney (2021); Carvalho et al. (2019)). Given the architectural constraints of the learner, it can, *e.g.*, be a post-hoc explanation method, but also an inherent explanation method if f is designed as such. Notably, this module is commonly also found in XIL (Friedrich et al., 2023) approaches. In summary, this module returns explanations, E_c, from the learner.

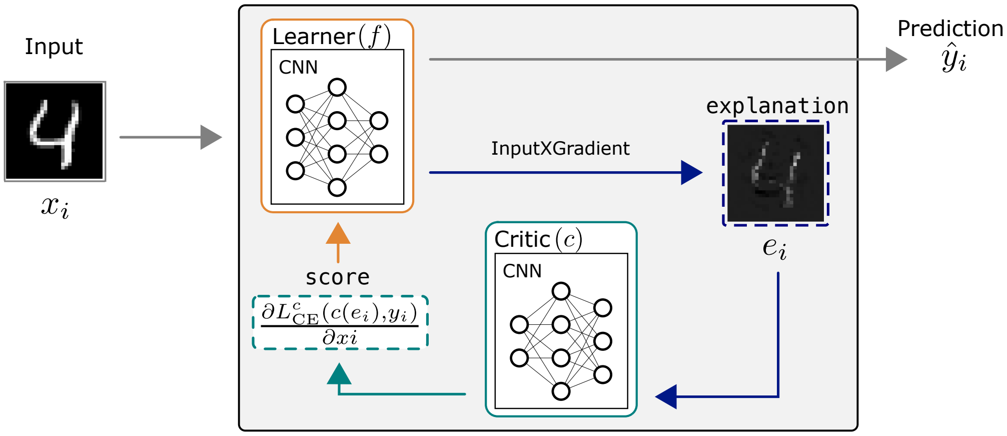

Figure 2: CNN-LSX: Learning by Self-Explaining instantiation for training CNNs for supervised image classification. Here CNNs represent both the *learner* and *critic*. Explanations are generated via InputXGradient. The `score` represents the classification loss signal of the critic on these explanations.

Reflect on explanations (Reflect). In the second core module, and arguably most distinctive module of LSX, the high-level role of the critic is to "reflect" on the quality of the explanations. This is an abstract measure and in LSX is quantified by the ability of the critic to perform the base task, given the learner's explanations. This explanation evaluation is performed in the Reflect module whereby the critic returns a `score` of the learner's `explanations`, given E_c and the corresponding original data that was used for generating these explanations, $\bar{X}_c$. In other words, `score` represents an indication of how well the critic performs the base task on the data $\bar{X}_c$ given the additional knowledge of the provided explanations, E_c (an idea that is related to (Pruthi et al., 2022)). What `score` exactly represents depends very much on the model type of the critic. For example, it can represent the signal from a classification loss function over $\bar{X}_c$ and E_c or a scoring of how probable an explanation is given the evidence of $\bar{X}_c$. By evaluating the quality of explanations based on their benefit in solving a task, the Reflect module represents one of the core aspects of LSX. Overall, explanations are thus treated not just as a one-way approach for indicating the importance of features when making a prediction (as often done in XAI). Rather, LSX specifically contributes to the view of explanations as verifiable rationales as in Fok & Weld (2023).

Integrate feedback on explanations (Revise). In the last module of an LSX loop, the feedback signal, `score`, obtained from the critic in the Reflect module, is used to refine the learner within Revise. This revision module can be realized in one of the many ways provided in interactive learning settings (Teso & Kersting, 2019; Friedrich et al., 2023). Standard revision tools entail loss-based methods (Ross et al., 2017; Selvaraju et al., 2019) as well as more parameter-efficient finetuning alternatives (Houlsby et al., 2019; Hu et al., 2022). But also non-differentiable revision approaches can be considered, *e.g.*, data augmentation approaches (Teso & Kersting, 2019) or retrieval-based setups that store the revisory feedback in an explicit revision engine (Friedrich et al., 2022; Tandon et al., 2022). In our LSX instantiations, we implement the revision step by adding an additional explanation loss, l_{expl} (*e.g.*, a HINT-like loss (Selvaraju et al., 2019) and a cross-entropy loss of the critic's classification) to the optimization process. The learner is thus jointly optimized in Revise via $L = l_{\text{B}} + \lambda l_{\text{expl}}$, where $\lambda \in \mathbb{R}$ represents a scaling factor.

3 Instantiating LSX

Many implementation details in the above overview depend strongly on the exact choice of the LSX components. Importantly, these are interdependent, *e.g.*, the choice of explanation method influences how the critic can evaluate the explanations, the choice of critic module influences the way the feedback score is computed, but also how this is integrated into the learner, etc. To give a better understanding of these abstract modules, in this section, we introduce two instantiations of how to integrate different base setups into LSX, where we refer to App. A.1 and A.2 for further details. These instantiations will also be the

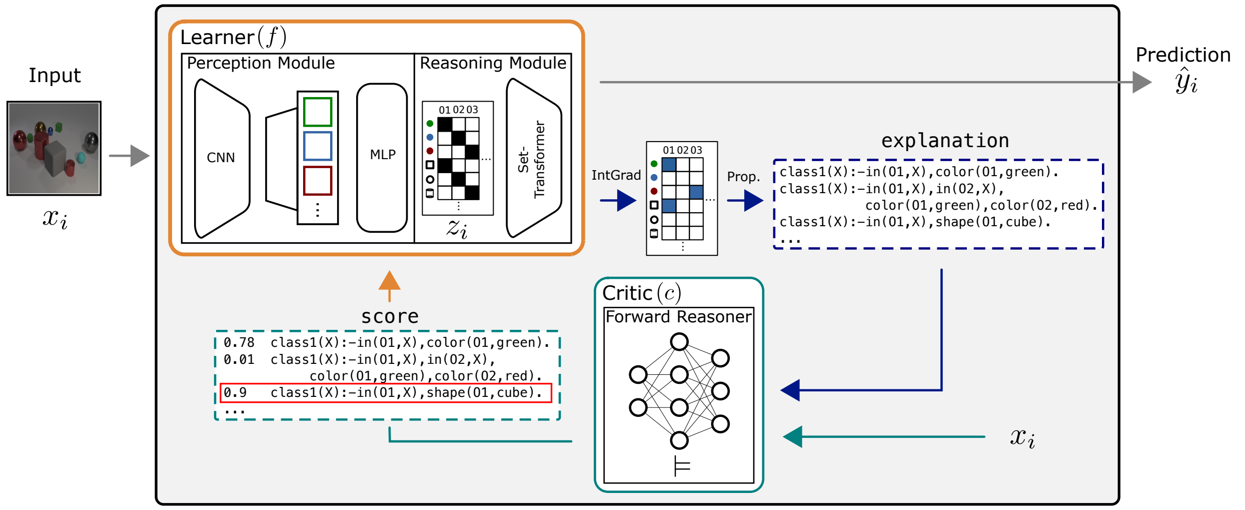
Input
x_i
Learner (f)
Perception Module
CNN
MLP
Reasoning Module
01 02 03
z_i
Set-Transformer
Prediction
$\hat{y}_i$
IntGrad
01 02 03
Prop.
explanation
class1(X):-in(01,X),color(01,green).
class1(X):-in(01,X),in(02,X),
color(01,green),color(02,red).
class1(X):-in(01,X),shape(01,cube).
...
score
0.78 class1(X):-in(01,X),color(01,green).
0.01 class1(X):-in(01,X),in(02,X),
color(01,green),color(02,red).
0.9 class1(X):-in(01,X),shape(01,cube).
...
Critic (c)
Forward Reasoner
$\models$
x_i

Figure 3: NeSyCL-LSX: Learning by Self-Explaining instantiation for supervised image classification via neuro-symbolic concept learner. The critic represents a neuro-symbolic forward reasoner, which computes all possible consequences in a differentiable manner given visual and logical input. An `explanation` represents a logical statement that is derived from concept-level saliency maps. The feedback `score` represents a probabilistic ranking of these logical explanations with which we identify the most likely explanation per image class and revise the learner to only produce this explanation.

point of investigation in our experimental evaluations. Both differ in several LSX components. Importantly, these two instantiations are by no means conclusive in how to integrate a base task into LSX, but serve as illustrations of how to integrate LSX into different models. We conclude this section with a more general perspective of Learning by Self-Explaining.

3.1 Neural and Neurosymbolic Instantiations

CNN-LSX via differentiable feedback score. We begin with a CNN-based setup shown in Fig. 2 (*cf.* App. A.1 for further details). This instantiation which we denote as CNN-LSX consists of a standard CNN as learner, f, and a duplicate of the learner as critic, c. Specifically, in this instantiation the learner is trained on raw images to predict the corresponding class labels and is optimized via a cross-entropy loss as $l_{\mathrm{B}} := l_{\mathrm{CE}}^{f}(f(X_f), y_f)$, thus representing the FIT module. The explain module is realized with the post-hoc, differentiable InputXGradient method (Shrikumar et al., 2017; Hechtlinger, 2016) and we compute $E_c = \text{EXPLAIN}(X_c|y_c)$ with $\bar{X}_c \subseteq \bar{X}_f$. As an `explanation` that is based on InputXGradient also contains the input, it is not required to separately pass X_c to the critic. In the REFLECT module, the critic therefore only receives these explanations as input and predicts the corresponding class labels from these $\hat{y}_c = c(E_c)$. The quality of the predictions is quantified via a second cross-entropy loss averaged over all samples in E_c, denoted as $l_{\mathrm{CE}}^{c}(c(E_c), y_c)$. Next, in REVISE, f is optimized via the joint loss $L = l_{\mathrm{CE}}^{f}(f(X_f), y_f) + \lambda l_{\mathrm{CE}}^{c}(c(E_c), y_c)$. In other words, the model parameters are updated based on both classification losses: the one from the learner given the training images X_f *and* that from the critic given the explanations of X_c. Lastly, we perform the final REVISE step (*i.e.*, when iteration budget T has been reached) with both loss functions evaluated on X_f: $L = l_{\mathrm{CE}}^{f}(f(X_f), y_f) + \lambda l_{\mathrm{CE}}^{c}(c(E_f), y_f)$.

NeSyCL-LSX via non-differentiable feedback score. Next, we will introduce an instantiation around the neuro-symbolic concept learner (NeSyCL) of Stammer et al. (2021) as base model (learner). We denote this instantiation as NeSyCL-LSX (*cf.* Fig. 3 and App. A.2 for further details). The NeSyCL consists of a slot attention-based perception module (Locatello et al., 2020) and set-transformer-based reasoning module (Lee et al., 2019). An image, x_i, is first processed by a pretrained perception module into a symbolic representation, $z_i \in [0,1]^{O \times A}$, indicating the presence of objects and their attributes in the corresponding image. Here, O indicates the number of objects (or slots) and A the number of predicted attributes. The reasoning module next makes a final class prediction given z_i. As in the CNN-LSX instantiation, NeSyCL-

LSX performs supervised image classification. This is the base task in the FIT module with l_{B} representing a cross-entropy loss. Obtaining `explanations` (EXPLAIN) is based on the approach of (Stammer et al., 2021) where saliency maps of z_i are created via Integrated Gradients (Sundararajan et al., 2017). These thus indicate which objects and which of their attributes are relevant for a prediction, denoted here as $e_i' \in [0,1]^{O \times A}$. By thresholding via an additional hyperparameter, $\delta \in [0,1]$, these saliency maps are binarized to $e_i'' \in \{0,1\}^{O \times A}$. We next propositionalize e_i'' by representing the explanation as a set of logical rules that are present in e_i''. These rules consist of conjunctive combinations of the important attributes and objects. We denote this set of candidate explanatory rules as e_i. Finally, by iterating over all samples in X_c and grouping the resulting candidate rules by image class we receive a set of candidate rules per class as $E_c = \{e^1, ..., e^K\}$. Moving on to the next modules, the critic in NeSyCL-LSX is represented by the neuro-symbolic forward reasoner of (Shindo et al., 2021; 2023), which computes all possible consequences in a differentiable manner given visual and logical input. In the REFLECT module the forward reasoner evaluates the `explanations`, E_c, by estimating the validity of each candidate explanation rule given the data in $\bar{X}_c$. Finally, based on the resulting estimated probabilities (`score`), the most probable rule per class j is chosen, denoted as e^j_{max}, and $E_{\text{max}} = \{e^1_{\text{max}}, ..., e^K_{\text{max}}\}$ is passed back to the learner (REVISE). Here, these logical explanations are next transformed back into binary vector form in the dimensions of the learner's original symbolic representation, $\bar{E}_{\text{max}} = \{\bar{e}^1_{\text{max}}, ..., \bar{e}^K_{\text{max}}\}$ with $\bar{e}^j_{\text{max}} \in \{0,1\}^{O \times A}$. Finally, f is updated via an additional mean-squared-error loss between the learner's explanations and these transformed explanations leading to $L = l^f_{\text{CE}}(f(X_f), y_f) + \lambda l_{\text{MSE}}(\bar{E}_{\text{max}}, E_f)$.

Configuration choices. As mentioned, many instantiations of LSX are possible, each with their own specific module and configuration choices. The instantiations introduced in this work, however, already sketch some interesting aspects and differences which we wish to highlight here. The most fundamental difference between CNN-LSX and NeSyCL-LSX lies within their REFLECT modules, specifically how a `score` of the learner's `explanations` are computed. In CNN-LSX the `score` represents a differentiable signal, where in NeSyCL-LSX this represents a probabilistic ranking of logical explanations. This second form of critiquing allows to weigh explanations and *e.g.*, identify the most "useful" explanation. The first form of critiquing, on the other hand, allows to fine-tune explanations. Related to this and concerning the EXPLAIN modules, where in CNN-LSX *continuous* input-level `explanations` are being processed, the logical `explanations` in NeSyCL-LSX are *discrete*. As an effect of this, the form of revision in the REVISE module differs. In CNN-LSX we can simply pass the backpropagated signal from the critic to the learner via a classification loss. In NeSyCL-LSX we identify the most-likely explanation. Additionally, in CNN-LSX the critic represents a duplicate CNN of its learner. In NeSyCL-LSX, on the other hand, the critic represents a different model type altogether compared to its learner.

3.2 General Perspective

In the following, we wish to provide a general perspective for LSX on important current challenges of AI.

Human-machine interactions. Accurate and trustworthy human-machine interactions have been identified as important criteria for the future deployability of AI systems (Friedrich et al., 2023; Teso et al., 2023; Holzinger, 2021; Angerschmid et al., 2022). Whereas much of ML research is not developed with this in mind the LSX framework automatically facilitates the development and integration of mechanisms that allow for fruitful human-machine interactions. *E.g.*, via the EXPLAIN module a human user can query the learner's reasons for a prediction and via the REVISE module integrate feedback on these explanations into the model. As explanations in LSX explicitly act as the means for revision and bidirectional communication between the two submodels, LSX fits well into the line of research that has identified the importance of leveraging explanations for bi-directional communication between human user and AI model (Teso et al., 2023). Finally, LSX does not ultimately remove the need for human-machine interactions for a sustainable model deployment, rather it can be used for an AI model to reflect on its learned reasons before receiving important human feedback. Deploying LSX in this way potentially reduces the amount and costs of human resources required (Friedrich et al., 2023).

System 1 and 2 processing. A prominent and well-studied hypothesis from cognitive psychology is that human cognition is mainly composed of two systems: an approximate, fast processing system (system 1) that handles the majority of familiar, daily situations and an embedded, slower, yet more exact system (system

2) that handles processing in more unfamiliar settings (Kahneman, 2011). This idea has recently gained much interest in the AI community (Goyal & Bengio, 2022; Kautz, 2022; Ganapini et al., 2022; Booch et al., 2021) and we here highlight several connections that can be made between the system 1 and 2 processing framework and Learning by Self-Explaining. Similar to the system 1 and 2 framework, within LSX there are two processing systems where one is embedded in the other. This differentiation is not necessarily based on the two submodels (*learner* and *critic*, *cf.* Fig. 1), but rather on the learning modules (*cf.* Alg. 1). Hereby, Fit takes over the fast, initial processing phase, and the triad consisting of Explain, Reflect and Revise results in a slower, embedded processing phase.

A major open question, particularly in AI research on system 1 and 2 processing, is what form of communication the two systems should be using (Goyal & Bengio, 2022; Kautz, 2022). In our work, we consider explanations to represent a valuable means of communication between the different systems. Specifically, the learner provides initial, associative explanations to its decisions which, in turn, are either directly interpreted as (*cf.* CNN-LSX) or transformed (*cf.* NeSyCL-LSX) into verifiable and refinable rationales. In line with Goyal & Bengio (2022), the process of explaining and reflecting on the learner's explanations can thus be seen as making the implicit knowledge of the learner explicit. At the same time, system 2 can also influence the processing of system 1. Evidence for this can be seen in our findings on explanation consolidation (*cf.* Tab. 3), where the critic's feedback leads to the learner adapting its explanations to represent more task-specific rationales. Lastly, relating to Henry Kautz's proposed Neuro[Symbolic] approach (Kautz, 2022) for AI system 1 and 2 processing, our NeSyCL-LSX has many parallels concerning the integration of neural and symbolic components. LSX in general, however, does not necessarily require a specific model type, but rather aims for an integration of any type of model via explanations.

Learning by self-explaining and causal inference. Another important viewpoint to consider is the connection between Learning by Self-Explaining and (the many different notions of) causal inference. Explanations in the general context of XAI and causality have been discussed in several recent works (Zečević et al., 2021; Heskes et al., 2020; Schwab & Karlen, 2019; Galhotra et al., 2021). However, in LSX they particularly play a role in querying and transforming implicit, associative knowledge into explicit knowledge. Through the explanation evaluation step in Reflect LSX is based on the idea that an explanation is "useful" if it represents a true rationale that, if applied, allows for an internal critic to perform a given task. This idea is related to the formalization of causal explanations by Woodward (2005) and picked up by Beckers (2022). In short, the main component of Woodward's formalization is that a "successful explanation" should provide us with information to perform interventional and counterfactual experiments. Lastly, the inherent idea of true rationale generation from explanation refinement found in LSX is closely related to findings of Fok & Weld (2023) which argue that the largest benefit of AI model explanations in hybrid decision making is given when the model's explanation allows the human decision maker to explicitly verify the model's prediction.

4 Experimental Evidence

In the following experiments, we investigate the benefits of Learning by Self-Explaining with the help of two instantiations, CNN-LSX and NeSyCL-LSX. Specifically, we compare the performances of LSX to the standard training setup (*i.e.*, supervised learning). Over the course of our evaluations, we will investigate the potential benefits of LSX concerning test-set generalization, explanation consolidation, explanation faithfulness and shortcut learning mitigation.

4.1 Experimental Setup

Data. We provide experimental results on four different datasets. Particularly, we evaluate CNN-LSX on the MNIST (LeCun et al., 1989) and ChestMNIST (Yang et al., 2023; Wang et al., 2017) datasets and NeSyCL-LSX on the concept-based datasets CLEVR-Hans3 (Stammer et al., 2021) and a variant of Caltech-UCSD Birds-200-2011 dataset (Wah et al., 2011), CUB-10 (*cf.* App. B). Overall, the number of training images in MNIST corresponds to 60k, in ChestMNIST to 78k, in CLEVR-Hans3 to 9k and in CUB-10 to 300 images. Finally, for investigating the effect of confounding factors as a form of shortcut learning (Geirhos et al., 2020), we evaluate accuracies on CLEVR-Hans3 and DecoyMNIST (Ross et al., 2017), a confounded variant of the MNIST dataset.

Table 1: Improved (few-shot) generalization via LSX on various datasets and models. We here present the accuracy in % on a held-out test set across varying training set sizes.

MNIST			
	1.2k	3k	full (60k)
CNN	89.83±0.2	93.83±0.08	**98.70**±0.1
CNN-LSX	**91.59**±0.91	**94.31**±0.43	98.03±0.2
ChestMNIST			
	1.6k	4k	full (78k)
CNN	58.68±0.15	58.49±0.31	60.86±0.08
CNN-LSX	**61.16**±0.54	**61.77**±0.75	**63.41**±1.3
CLEVR-Hans3			
	150	375	full (9k)
NeSyCL	91.40±1.80	96.81±0.94	99.00±0.28
NeSyCL-LSX	**94.51**±1.94	**97.34**±0.44	**99.08**±0.17
CUB-10			
	100	150	full (300)
NeSyCL	83.57±1.67	87.14±0.4	93.13±0.4
NeSyCL-LSX	**84.49**±1.18	**93.05**±1.72	**96.33**±0.31
avg. improvement	**2.07**	**2.55**	**1.29**

Table 2: Mitigating confounders via LSX: Test set performances on confounded datasets, both with deconfounded samples during training (*w/ deconf.*) and without (*w/ conf.*).

DecoyMNIST		
	w/ conf.	w/ deconf.
CNN	63.52±1.39	86.88±0.68
CNN-LSX	**78.99**±2.71	**88.43**±2.34
CLEVR-Hans3		
	w/ conf.	w/ deconf.
NeSyCL	85.96±4.20	91.23±1.2
NeSyCL-LSX	**90.90**±4.38	**95.64**±2.21

Metrics. We provide evaluations of LSX based on five metrics where we briefly describe these here and refer to App. C for details. **(1)** The first metric is the standard *balanced classification accuracy* on a held-out test set. **(2)** For investigating the revised explanations via LSX we provide the classification accuracy of a linear, ridge regression model. This model is optimized to classify a set of the learner's explanations, given their corresponding ground-truth class labels and evaluated on a held-out set of explanations. **(3)** We further provide a cluster analysis based metric over all explanations, similar to the Dunn index (Dunn, 1973; 1974). This metric, which we denote as *Inter- vs. Intraclass Explanation Similarity* (IIES), quantifies how similar explanations are within one class, but dissimilar between classes (lower values indicate better separability). For investigating whether the learner in fact makes a decision based on the reported explanations, we analyse the faithfulness (Hooker et al., 2019; Chan et al., 2022) of the learner's explanations via two metrics as introduced by (DeYoung et al., 2020), namely **(4)** *sufficiency* and **(5)** *comprehensiveness*. Both metrics measure the impact on the model's performances of removing specific parts of the input based on the explanations. For comprehensiveness, parts of the input are removed which correspond to important features as identified by the explanation. For sufficiency, parts of the input are removed which correspond to unimportant features as identified by the explanation. For continuous input settings (MNIST and ChestMNIST) we modify the computation of these two metrics slightly by subtracting the impact when randomly chosen features are removed. This way we compensate for the potential influence of such out-of-distribution input. Notably, this can lead to negative values. In both formulations, however, higher comprehensiveness and lower sufficiency scores are better. Both metrics are not normalized and provide a relative comparison.

Setup. In all evaluations, we compare the performances of the LSX instantiations with the performances of the base learners that were trained for the same overall number of epochs, but only in the standard supervised manner. These are denoted as CNN and NeSyCL. We evaluate CNN-LSX on the MNIST and ChestMNIST datasets and NeSyCL-LSX on the CLEVR-Hans3 and CUB-10 datasets. The baselines were trained on the same data as the LSX versions, *i.e.*, $\bar{X}_f^{\text{baseline}} = \bar{X}_f^{\text{LSX}} \cup \bar{X}_c^{\text{LSX}}$. Note that for NeSyCL-LSX for CUB-10, we replaced the slot-attention perception module with a pretrained Inception-v3 network (Szegedy et al., 2016) and the reasoning module with a single linear layer as in (Koh et al., 2020). We provide all results as mean values with standard deviations over five runs with different random seeds. Lastly, for investigating shortcut behavior we provide balanced accuracy scores on the unconfounded held-out test sets of Decoy-MNIST and CLEVR-Hans3 while being trained on the confounded train set. In the CLEVR-Hans3 evaluations that do not target evaluating the effect of confounding factors, the original confounded evaluation set of CLEVR-Hans3 was used as held-out test set.

Table 3: Explanation consolidation via LSX. The metrics here are Inter- vs. Intraclass Explanation Similarity (IIES) of a learner's explanations (left) and the classification accuracy of a ridge regression model (RR. Acc., in %) on the learner's explanations (right). Both metrics are proxies for the explanation similarity within a class, yet separability between classes.

	IIES (↓)	RR Acc. (↑)
	MNIST	
CNN	2.7±0.07	12.32±0.35
CNN-LSX	**0.7**±0.01	**99.91**±0.06
	ChestMNIST	
CNN	3.89±0.13	74.87±0.24
CNN-LSX	**0.75**±0.05	**99.92**±0.03
	CLEVR-Hans3	
NeSyCL	0.65±0.07	93.48±2.41
NeSyCL-LSX	**0.2**±0.06	**100**±0.0
	CUB-10	
NeSyCL	0.0266±0.0005	**100**±0.0
NeSyCL-LSX	**0.0024**±0.0001	**100**±0.0

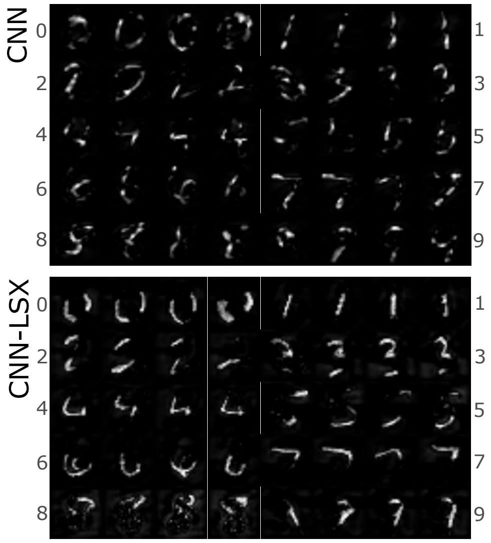

Figure 4: Exemplary explanations on MNIST of CNN baseline vs. CNN-LSX. Four random explanations are shown per image class (class ids on sides).

4.2 Experimental Results

Improved (few-shot) generalisation. An intuitive aspect of psychological findings on learning via self-explaining is that reflecting on one's learned explanations leads to improved generalizable knowledge (Chi et al., 1994). We investigate LSX in our first evaluation by measuring the held-out test set accuracy of CNN-LSX on the MNIST and ChestMNIST datasets and of NeSyCL-LSX on the CLEVR-Hans3 and CUB-10 datasets. In the rightmost column of Tab. 1, we present the respective test set accuracies for all learning configurations when trained on the full training size of each dataset. We can observe that on average, *i.e.*, over all datasets and over both LSX instantiations, there is a substantial boost in test set accuracy (last row). In the remaining columns of Tab. 1, we present the test-set accuracy in the smaller-data regime, *i.e.*, when the models were trained on different-sized subsets of the original training set [2]. We can observe large performance gains with LSX for all configurations and datasets. Particularly, these improvements are on average greater than those observed on the full training set sizes. Altogether these results suggest that learning via self-explaining leads to improved test-set generalization performances with larger improvements, particularly in small-data regimes.

Self-unconfounding. In the second set of evaluations, we are interested in how far LSX can help mitigate shortcut behaviour (Geirhos et al., 2020). We particularly focus on confounded behaviour as a form of shortcut learning in which a learner picks up on spurious correlations within the training dataset that are not present in the test set (Schramowski et al., 2020). We investigate two settings. (i) With the first (denoted as *w/ conf.*), we investigate the performance of LSX in mitigating confounding behaviour without additional knowledge on the confounding factors. To this end, we train the two LSX instantiations (and baselines) as in the previous setups with $\bar{X}_c \subseteq \bar{X}_f$, and $\bar{X}_f$ representing a confounded dataset. (ii) In the second setting, $\bar{X}_c$ represents a dataset that is with-held from training the learner (*i.e.*, $\bar{X}_c \cap \bar{X}_f = \emptyset$) and represents a dataset that is explicitly deconfounded (*cf.* App. C.3 for details on the sizes of $\bar{X}_c$). In other words, the spurious correlation found in X_f is not present in $\bar{X}_c$. We denote this case as *w/ deconf.* For the standard training scheme the models have access to $\bar{X}_c$ within their training phase. We evaluate the CNN-LSX configuration on the Decoy-MNIST dataset and the NeSyCL-LSX configuration on the CLEVR-Hans dataset.

---

[2] Hereby, the size of these subsets varies over the different datasets due to different specifics of each dataset, e.g. the original training set sizes.

Table 4: Explanation faithfulness via LSX: Comprehensiveness and sufficiency results of explanations for models trained on all training samples.

	Comp. (↑)	Suff. (↓)
	MNIST	
CNN	−1.34±0.39	23.11±1.18
CNN-LSX	**16.49**±2.79	**-0.21**±4.18
	ChestMNIST	
CNN	13.98±0.43	−4.2±1.84
CNN-LSX	**18.84**±0.38	**-8.55**±1.92

	Comp. (↑)	Suff. (↓)
	CLEVR-Hans3	
NeSyCL	59.3±3.6	21.67±4.55
NeSyCL-LSX	**63.26**±2.57	**7.73**±1.28
	CUB-10	
NeSyCL	64.54±0.2	10.44±0.45
NeSyCL-LSX	**64.59**±0.23	**6.3**±0.34

In Tab. 2, we present the held-out test set accuracies of all configurations. We observe a strong improvement in performances when training LSX on the deconfounded critic sets (*w/ deconf.*), indicating that reflecting on the explanations of an explicitly deconfounded critic set can lead to much improved shortcut mitigation behaviour compared to the baseline learning setup. Even more interesting is the result of the *w/ conf.* setting. We observe that the LSX trained models, though never having seen deconfounded data, lead to strong mitigation improvements. This result suggests great practical implications, as it does not require prior knowledge on the confounding factors. Overall, our results suggest a large beneficial effect of Learning by Self-Explaining in mitigating the issue of shortcut learning, specifically confounded behaviour.

Explanation consolidation. In the next evaluation, we wish to analyze how the critic's feedback signal influences the learner's representations, specifically its explanations. Based on the intuition behind the Reflect module concerning "good" explanations, we hypothesize that the explanations of a LSX trained model represent more task-specific rationales. We present our results based on the Inter- vs. Intraclass Explanation Similarity (IIES) and the accuracy of a ridge regression model that was trained to classify a set of explanations and evaluated on a second, held-out set. Both of these metrics measure the class-based separability of a model's explanations. In Tab. 3, one can observe that over both metrics training via LSX leads to much more separable and distinct explanations. This effect appears less pronounced for the NeSy datasets, which is likely due to the sparseness and low dimensionality of their concept-level data and therefore also of the explanations. In Fig. 4, we also provide qualitative results of the explanation consolidation for MNIST, where explanations from four randomly sampled input samples are presented for each digit class. These visualizations undermine the quantitative results of Tab. 3 and particularly indicate the distinctness of the explanations from a LSX trained model within one data class from those of other classes. Overall, we observe that LSX leads to more consistent explanations across samples of one class, yet distinctly separate explanations to samples of other classes. We refer to such an effect as *explanation consolidation*.

Explanation faithfulness. Although the performance improvements of the first evaluation suggest that LSX learners do make use of the critic's *explanatory feedback*, and the evaluations regarding explanation consolidation indicate that models learn more distinct explanations via LSX, an open question remains whether the learner's in fact make use of these *explanations* for making their decisions. In other words: are a learner's explanations faithful to its decision process? This is a relevant question on its own, particularly in the field of XAI (Hooker et al., 2019; DeYoung et al., 2020) and XIL (Schramowski et al., 2020) as models that produce unfaithful explanations are detrimental for building trust between human users and machines and at the same time make potential revisions via these explanations difficult (Schramowski et al., 2020; Teso et al., 2023). To investigate the faithfulness of LSX-learned explanations we turn to established faithfulness metrics of AI literature. Specifically, we use the *sufficiency* and *comprehensiveness* metrics of DeYoung et al. (2020). In Tab. 4, we present the results of these metrics over all four datasets and both LSX implementations[3]. One can observe a strong improvement via LSX in both metrics across all models and datasets. Specifically, the comprehensiveness results indicate that the information, considered as relevant by the explanations learned via LSX, are indeed important for the model to make its prediction. At the same time, the sufficiency results indicate that less important information based on the explanations has

---

[3] For CNN-LSX, we adapt these for handling continuous data (*cf.* App. C).

a decreased impact on the learner's decisions. Overall, we conclude that indeed training via LSX leads to more faithful explanations.

Overall, our evaluations demonstrate the benefits of training via LSX on a variety of important tasks and metrics that go beyond standard evaluations of ML research.

5 Related Works

LSX is related to work in explainable AI (XAI), leveraging explanations in ML and, importantly, model refinement via self-refinement or feedback from a second model. Let us highlight these works in the following.

5.1 (Leveraging) Explanations in ML

Receiving explanations to an AI model's decision has become a heavily advocated and researched topic in recent years, culminating in the field of *explainable AI* (XAI) (*cf.* Guidotti et al. (2019); Ras et al. (2022a); Roy et al. (2022); Saeed & Omlin (2023) for valuable overviews) and *interpretable AI*, which focuses on developing models that are *explicitly* interpretable by design (Räuker et al., 2023; Li et al., 2018; Rudin et al., 2022; Rudin, 2019). An additional branch of research can be placed between explainable AI and interpretable AI, namely that of *self-explaining models* (Alvarez-Melis & Jaakkola, 2018; Lee et al., 2022; Roy et al., 2022; Camburu et al., 2018; Bastings et al., 2019; Majumder et al., 2022). In all of these works and in contrast to LSX, explanations are only provided in a one-way communication as a means of model inspection for humans and not considered as a means of model refinement.

The idea of leveraging explanations in the training process has only recently been picked up by parts of the ML community. In the field of explanatory interactive learning (XIL) (Teso & Kersting, 2019; Schramowski et al., 2020; Stammer et al., 2021; Friedrich et al., 2023) human users provide revisory feedback on the explanations of an ML model. Similar ideas can also be identified in other works of human-machine interactive learning (Teso et al., 2023; Gao et al., 2022), *e.g.*, in preference selection based interactions for learning vision language models (Brack et al., 2022). Compared to these, we argue for the importance of leveraging explanations in the training loop even before the necessity of human-machine interactions and advocate for the potential of explanations in a form of self-refinement in a model's initial learning process.

In contrast, several works have identified the value of leveraging explanations outside of human-interactive learning (*e.g.*, (Giunchiglia et al., 2022; Lampinen et al., 2021; 2022; Norelli et al., 2022)). In the works of Lei et al. (2016) and Bastings et al. (2019) (later categorized under the term *explain-then-predict models* by Zhang et al. (2021)), the goal is for one model to learn to extract the rationale[4] from an input and a second model to learn to predict the final class from these rationales. Similar ideas were picked up by (Zhang et al., 2021; Krishna et al., 2023). None of these works evaluate the correctness of explanations and particularly none use explanations as a means to *revise* a model.

5.2 (Self-)Refinement in ML

A recent, but quickly growing field of research related to our work is that which we categorize under the term of *self-refining AI*. This roughly encompasses research that investigates forms of self-supervised refinement of an AI model, *e.g.*, Wang et al. (2023) propose an approach for instruction-tuning. In the self-alignment approach of Sun et al. (2023), a LLM is aligned with few human provided principles. Schick et al. (2021), on the other hand, identify that LLMs can, to a certain degree, identify biases in their own generations and the authors leverage this characteristic in a finetuning process to mitigate biased generation in future prompts. In the work of Madaan et al. (2023) a model is used to provide feedback to its initial generations, where the feedback generation is guided via targeted, few-shot prompting. Zelikman et al. (2022), on the other hand, investigate finetuning a model by based on generated "chain-of-thought" rationales that lead to correct task predictions. Lastly, Paul et al. (2023) propose an approach in which a model learns to provide explicit intermediate reasoning steps for an answer via feedback from a critic model. Importantly in this work, the critic is specifically trained to identify false reasoning steps. In contrast to LSX only few of these mentioned

---

[4] Here we mean the term "rationale" as adopted in research on explainability in NLP.

approaches focus on refinement via explanations. Those that do require specifically trained modules for providing feedback on the explanations. In contrast in LSX explanations are quantified in how far they can help perform a task. Thus the evaluation and refinement of a model is performed without specific pretraining or prompt specification.

In contrast to self-refining AI a different branch of research focuses on revising a model based on forms of feedback from a second model. Such et al. (2020) which represents a meta-learning training data generation process in which a data generator and learner model are optimized for the same goal of improving the learner's performance on a given task. Nair et al. (2023) propose a general chat framework that leverages two agents, *researcher* and *decider*, to iteratively work through a task. The researcher plays the role of making task-specific suggestions to the decider, where the decider responds to the information provided by the researcher. In the student-teacher framework (Wang & Yoon, 2022) the goal is knowledge distillation, *i.e.*, learned knowledge from a trained model should be conveyed to a second model, the student model. Somewhat related to this is the concept of self-paced learning within the field of curriculum learning (Kumar et al., 2010; Wang et al., 2022) in which a model provides a signal on how "fast" to learn. Interestingly, Pruthi et al. (2022) frame the utility of an explanation in a student-teacher setup in which the goal is for a student model to simulate a teacher's behaviour best possible. Also Schneider & Vlachos (2023) argue for the importance of explanations in reflective processes. However, the authors only propose an approach where a model makes a final prediction based on the input and explanation that is estimated by a second model, similar to (Lei et al., 2016; Bastings et al., 2019; Zhang et al., 2021; Krishna et al., 2023). Overall, these approaches have a different target and motivation than our work. Particularly, in LSX the role of the critic submodel is to represent an internal optimization loop based on whether the explanations provided from the learner are beneficial in performing the original task.

6 Conclusion

In this work, we have introduced a novel learning framework, Learning by Self-Explaining (LSX), with which we argue for a novel perspective on the role of self-explaining in the process of learning in AI models. Thus, we claim that explanations are important not just for human users to understand or revise an AI model, but that they can play an even more important role in a form of self-reflection in which an agent assesses its own learned knowledge via its explanations. Our experimental evaluations highlight several benefits of training via LSX in the context of generalization, knowledge consolidation, explanation faithfulness and shortcut mitigation. Conclusively, with this work, we argue and provide evidence for the potential of explanations within a models learning process and, ultimately, for developing *reflective* AI.

Despite the promising results of our instantiations there is still great potential for other design choices. Investigating such instantiations and their benefits is an essential avenue for future research. *E.g.*, applying LSX to settings with other modalities, such as natural language, but also with several modalities, *i.e.*, based on current multimodal models (Radford et al., 2021; Li et al., 2022; Rombach et al., 2021; Alayrac et al., 2022). A more conceptual direction is the integration of a memory buffer of past LSX optimized explanations, allowing for models to re-iterate over previous generations of explanations (Chi et al., 1994). Another interesting point lies in adding apriori constraints onto the learner's explanations, *e.g.*, by integrating background knowledge into the explanation reflection process. In this way, explanations can be assessed not just based on the critic's usability for performing the base task, but also based on the agreement of the true rationale with verified background knowledge. Related to this and to the discussion on the system 1 and 2 processing framework, future research should investigate the potential of LSX *e.g.*, via a memory of explanations for detecting and handling out-of-distribution situations (Goyal & Bengio, 2022). As a first step, in the formulation of LSX and in our specific instantiations we have focused on supervised learning as base task. Another crucial avenue going forward is to apply LSX to other forms of supervision, such as self-supervised learning or reinforcement learning approaches *e.g.*, integrating into actor-critic approaches or for guiding curiosity driven replay (Kauvar et al., 2023).

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning. In *Advances on Neural Information Processing Systems (NeurIPS)*, 2022.

David Alvarez-Melis and Tommi S. Jaakkola. Towards robust interpretability with self-explaining neural networks. pp. 7786–7795, 2018.

Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. Towards better understanding of gradient-based attribution methods for deep neural networks. In *International Conference on Learning Representations (ICLR)*. OpenReview.net, 2018.

Alessa Angerschmid, Jianlong Zhou, Kevin Theuermann, Fang Chen, and Andreas Holzinger. Fairness and explanation in ai-informed decision making. *Machine Learning and Knowledge Extraction*, 4(2):556–579, 2022.

Jasmijn Bastings, Wilker Aziz, and Ivan Titov. Interpretable neural predictions with differentiable binary variables. In Anna Korhonen, David R. Traum, and Lluís Màrquez (eds.), *Conference of the Association for Computational Linguistics (ACL)*, pp. 2963–2977. Association for Computational Linguistics, 2019.

Sander Beckers. Causal explanations and XAI. In Bernhard Schölkopf, Caroline Uhler, and Kun Zhang (eds.), *Conference on Causal Learning and Reasoning (CLeaR)*, volume 177 of *Proceedings of Machine Learning Research*, pp. 90–109. PMLR, 2022.

Anna Belobrovy. Theories on self-reflection in education. In *Asian Conference on Education*, 2018.

Kiran Bisra, Qing Liu, John C Nesbit, Farimah Salimi, and Philip H Winne. Inducing self-explanation: A meta-analysis. *Educational Psychology Review*, 30:703–725, 2018.

Grady Booch, Francesco Fabiano, Lior Horesh, Kiran Kate, Jonathan Lenchner, Nick Linck, Andrea Loreggia, Keerthiram Murugesan, Nicholas Mattei, Francesca Rossi, and Biplav Srivastava. Thinking fast and slow in AI. In *Conference on Artificial Intelligence (AAAI)*, pp. 15042–15046. AAAI Press, 2021.

Manuel Brack, Patrick Schramowski, Björn Deiseroth, and Kristian Kersting. Illume: Rationalizing vision-language models by interacting with their jabber. *CoRR*, 2022.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. e-snli: Natural language inference with natural language explanations. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 9560–9572, 2018.

Diogo V Carvalho, Eduardo M Pereira, and Jaime S Cardoso. Machine Learning Interpretability: A Survey on Methods and Metrics. *Electronics*, 8(8):832, 2019.

Martine Chamberland and Sílvia Mamede. Self-explanation, an instructional strategy to foster clinical reasoning in medical students. *Health Professions Education*, 1(1):24–33, 2015.

Chun Sik Chan, Huanqi Kong, and Guanqing Liang. A comparative study of faithfulness metrics for model interpretability methods. In *Conference of the Association for Computational Linguistics (ACL)*, pp. 5029–5038. Association for Computational Linguistics, 2022.

Michelene TH Chi. Learning from examples via self-explanations. In *Knowing, learning, and instruction*, pp. 251–282. Routledge, 2018.

Michelene TH Chi, Paul J Feltovich, and Robert Glaser. Categorization and representation of physics problems by experts and novices. *Cognitive science*, 5(2):121–152, 1981.

Michelene TH Chi, Nicholas De Leeuw, Mei-Hung Chiu, and Christian LaVancher. Eliciting self-explanations improves understanding. *Cognitive science*, 18(3):439–477, 1994.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. ERASER: A benchmark to evaluate rationalized NLP models. In *Conference of the Association for Computational Linguistics (ACL)*, pp. 4443–4458. Association for Computational Linguistics, 2020.

Joseph C Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. 1973.

Joseph C Dunn. Well-separated clusters and optimal fuzzy partitions. *Journal of cybernetics*, 4(1):95–104, 1974.

Shmuel Ellis, Bernd Carette, Frederik Anseel, and Filip Lievens. Systematic reflection: Implications for learning from failures and successes. *Current Directions in Psychological Science*, 23(1):67–72, 2014.

Raymond Fok and Daniel S. Weld. In search of verifiability: Explanations rarely enable complementary performance in ai-advised decision making. *CoRR*, abs/2305.07722, 2023.

Felix Friedrich, Wolfgang Stammer, Patrick Schramowski, and Kristian Kersting. Revision transformers: Getting rit of no-nos. *CoRR*, abs/2210.10332, 2022.

Felix Friedrich, Wolfgang Stammer, Patrick Schramowski, and Kristian Kersting. A typology for exploring the mitigation of shortcut behaviour. *Nature Machine Intelligence*, 5(3):319–330, 2023.

Sainyam Galhotra, Romila Pradhan, and Babak Salimi. Explaining black-box algorithms using probabilistic contrastive counterfactuals. In *International Conference on Management of Data (SIGMOD)*, pp. 577–590. ACM, 2021.

Marianna Bergamaschi Ganapini, Murray Campbell, Francesco Fabiano, Lior Horesh, Jon Lenchner, Andrea Loreggia, Nicholas Mattei, Francesca Rossi, Biplav Srivastava, and Kristen Brent Venable. Thinking fast and slow in AI: the role of metacognition. In *Machine Learning, Optimization, and Data Science - 8th International Workshop (LOD)*, volume 13811 of *Lecture Notes in Computer Science*, pp. 502–509. Springer, 2022.

Yuyang Gao, Siyi Gu, Junji Jiang, Sungsoo Ray Hong, Dazhou Yu, and Liang Zhao. Going beyond XAI: A systematic survey for explanation-guided learning. *CoRR*, abs/2212.03954, 2022.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard S. Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

Valentina Giunchiglia, Chirag Varun Shukla, Guadalupe Gonzalez, and Chirag Agarwal. Towards training gnns using explanation directed message passing. In Bastian Rieck and Razvan Pascanu (eds.), *Learning on Graphs Conference (LoG)*, volume 198 of *Proceedings of Machine Learning Research*, pp. 28. PMLR, 2022.

Michaela Gläser-Zikuda. *Self-Reflecting Methods of Learning Research*, pp. 3011–3015. Springer US, 2012.

Anirudh Goyal and Yoshua Bengio. Inductive biases for deep learning of higher-level cognition. *Proceedings of the Royal Society A*, 2022.

Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A Survey of Methods for Explaining Black Box Models. *ACM Computing Surveys (CSUR)*, 51(5):1–42, 2018.

Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM Computing Surveys*, 51(5):93:1–93:42, 2019.

Yotam Hechtlinger. Interpretation of prediction models using the input gradient. *CoRR*, abs/1611.07634, 2016.

Tom Heskes, Evi Sijben, Ioan Gabriel Bucur, and Tom Claassen. Causal shapley values: Exploiting causal knowledge to explain individual predictions of complex models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Andreas Holzinger. The next frontier: AI we can really trust. In *Machine Learning and Principles and Practice of Knowledge Discovery in Databases - International Workshops of ECML PKDD*, volume 1524 of *Communications in Computer and Information Science*, pp. 427–440. Springer, 2021.

Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. A benchmark for interpretability methods in deep neural networks. In *Conference on Neural Information Processing Systems (NeurIPS 2019)*, pp. 9734–9745, 2019.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning (ICML)*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2790–2799, 2019.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

Daniel Kahneman. *Thinking, fast and slow.* Macmillan, 2011.

Henry Kautz. The third ai summer: Aaai robert s. engelmore memorial lecture. *AI Magazine*, 43(1):93–104, 2022.

Isaac Kauvar, Chris Doyle, Linqi Zhou, and Nick Haber. Curious replay for model-based adaptation. *CoRR*, abs/2306.15934, 2023.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International Conference on Machine Learning (ICML)*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5338–5348. PMLR, 2020.

Satyapriya Krishna, Jiaqi Ma, Dylan Slack, Asma Ghandeharioun, Sameer Singh, and Himabindu Lakkaraju. Post hoc explanations of language models can improve language models. *CoRR*, abs/2305.11426, 2023.

M. Pawan Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems (NeurIPs)*, pp. 1189–1197. Curran Associates, Inc., 2010.

Kyungbin Kwon and David H Jonassen. The influence of reflective self-explanations on problem-solving performance. *Journal of Educational Computing Research*, 44(3):247–263, 2011.

Andrew K. Lampinen, Nicholas A. Roy, Ishita Dasgupta, Stephanie C. Y. Chan, Allison C. Tam, James L. McClelland, Chen Yan, Adam Santoro, Neil C. Rabinowitz, Jane X. Wang, and Felix Hill. Tell me why! - explanations support learning of relational and causal structure. *CoRR*, abs/2112.03753, 2021.

Andrew K. Lampinen, Ishita Dasgupta, Stephanie C. Y. Chan, Kory Matthewson, Michael Henry Tessler, Antonia Creswell, James L. McClelland, Jane X. Wang, and Felix Hill. Can language models learn from explanations in context? *CoRR*, abs/2204.02329, 2022.

Douglas P Larsen, Andrew C Butler, and Henry L Roediger III. Comparative effects of test-enhanced learning and self-explanation on long-term retention. *Medical education*, 47(7):674–682, 2013.

Yann LeCun, Bernhard Boser, John Denker, Donnie Henderson, Richard Howard, Wayne Hubbard, and Lawrence Jackel. Handwritten digit recognition with a back-propagation network. *Advances on Neural Information Processing Systems (NeurIPS)*, 2, 1989.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam R. Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International Conference on Machine Learning (ICML)*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3744–3753. PMLR, 2019.

Seungeon Lee, Xiting Wang, Sungwon Han, Xiaoyuan Yi, Xing Xie, and Meeyoung Cha. Self-explaining deep models with logic rule reasoning. 2022.

Tao Lei, Regina Barzilay, and Tommi S. Jaakkola. Rationalizing neural predictions. In Jian Su, Xavier Carreras, and Kevin Duh (eds.), *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 107–117. The Association for Computational Linguistics, 2016.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven C. H. Hoi. Blip: Bootstrapping language-image pretraining for unified vision-language understanding and generation. In *International Conference on Machine Learning (ICML)*, 2022.

Oscar Li, Hao Liu, Chaofan Chen, and Cynthia Rudin. Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. In *Conference on Artificial Intelligence (AAAI)*, pp. 3530–3537. AAAI Press, 2018.

Q Vera Liao and Kush R Varshney. Human-centered explainable ai (xai): From algorithms to user experiences. *CoRR*, abs/2110.10790, 2021.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *Advances on Neural Information Processing Systems (NeurIPS)*, 2020.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Sean Welleck, Bodhisattwa Prasad Majumder, Shashank Gupta, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. *CoRR*, abs/2303.17651, 2023.

Bodhisattwa Prasad Majumder, Oana Camburu, Thomas Lukasiewicz, and Julian J. McAuley. Knowledge-grounded self-rationalization via extractive and natural language explanations. volume 162, pp. 14786–14801. PMLR, 2022.

Varun Nair, Elliot Schumacher, Geoffrey J. Tso, and Anitha Kannan. DERA: enhancing large language model completions with dialog-enabled resolving agents. *CoRR*, abs/2303.17071, 2023.

Antonio Norelli, Giorgio Mariani, Luca Moschella, Andrea Santilli, Giambattista Parascandolo, Simone Melzi, and Emanuele Rodolà. Explanatory learning: Beyond empiricism in neural networks. *CoRR*, abs/2201.10222, 2022.

Liangming Pan, Michael Saxon, Wenda Xu, Deepak Nathani, Xinyi Wang, and William Yang Wang. Automatically correcting large language models: Surveying the landscape of diverse self-correction strategies. *CoRR*, abs/2308.03188, 2023.

Debjit Paul, Mete Ismayilzada, Maxime Peyrard, Beatriz Borges, Antoine Bosselut, Robert West, and Boi Faltings. REFINER: reasoning feedback on intermediate representations. *CoRR*, abs/2304.01904, 2023.

Danish Pruthi, Rachit Bansal, Bhuwan Dhingra, Livio Baldini Soares, Michael Collins, Zachary C. Lipton, Graham Neubig, and William W. Cohen. Evaluating explanations: How much do explanations from the teacher aid students? *Transactions of the Association for Computational Linguistics*, 10:359–375, 2022.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021.

Gabrielle Ras, Ning Xie, Marcel van Gerven, and Derek Doran. Explainable deep learning: A field guide for the uninitiated. *Journal of Artificial Intelligence Research*, 73:329–396, 2022a.

Gabrielle Ras, Ning Xie, Marcel van Gerven, and Derek Doran. Explainable deep learning: A field guide for the uninitiated. *Journal of Artificial Intelligence Research*, 73:329–397, 2022b.

Tilman Räuker, Anson Ho, Stephen Casper, and Dylan Hadfield-Menell. Toward transparent ai: A survey on interpreting the inner structures of deep neural networks. In *IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pp. 464–483. IEEE, 2023.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Computer Vision and Pattern Recognition Conference (CVPR)*, 2021.

Andrew Slavin Ross, Michael C. Hughes, and Finale Doshi-Velez. Right for the right reasons: Training differentiable models by constraining their explanations. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2662–2670, 2017.

Nicholas A. Roy, Junkyung Kim, and Neil C. Rabinowitz. Explainability via causal self-talk. 2022.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.

Cynthia Rudin, Chaofan Chen, Zhi Chen, Haiyang Huang, Lesia Semenova, and Chudi Zhong. Interpretable machine learning: Fundamental principles and 10 grand challenges. *Statistic Surveys*, 16:1–85, 2022.

Waddah Saeed and Christian W. Omlin. Explainable AI (XAI): A systematic meta-survey of current challenges and future opportunities. *Knowledge-Based Systems*, 263:110273, 2023.

Timo Schick, Sahana Udupa, and Hinrich Schütze. Self-diagnosis and self-debiasing: A proposal for reducing corpus-based bias in NLP. *Transactions of the Association for Computational Linguistics*, 9:1408–1424, 2021.

Johannes Schneider and Michalis Vlachos. Reflective-net: Learning from explanations. *Data Mining and Knowledge Discovery*, pp. 1–22, 2023.

Patrick Schramowski, Wolfgang Stammer, Stefano Teso, Anna Brugger, Franziska Herbert, Xiaoting Shao, Hans-Georg Luigs, Anne-Katrin Mahlein, and Kristian Kersting. Making deep neural networks right for the right scientific reasons by interacting with their explanations. *Nature Machine Intelligence*, 2(8): 476–486, 2020.

Patrick Schwab and Walter Karlen. Cxplain: Causal explanations for model interpretation under uncertainty. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 10220–10230, 2019.

Ramprasaath Ramasamy Selvaraju, Stefan Lee, Yilin Shen, Hongxia Jin, Shalini Ghosh, Larry P. Heck, Dhruv Batra, and Devi Parikh. Taking a HINT: leveraging explanations to make vision and language models more grounded. In *International Conference on Computer Vision (ICCV)*, pp. 2591–2600. IEEE, 2019.

Hikaru Shindo, Devendra Singh Dhami, and Kristian Kersting. Neuro-symbolic forward reasoning. *CoRR*, abs/2110.09383, 2021.

Hikaru Shindo, Viktor Pfanschilling, Devendra Singh Dhami, and Kristian Kersting. αilp: thinking visual scenes as differentiable logic programs. *Machine Learning*, 112(5):1465–1497, 2023.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International Conference on Machine Learning (ICML)*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3145–3153. PMLR, 2017.

Wolfgang Stammer, Patrick Schramowski, and Kristian Kersting. Right for the right concept: Revising neuro-symbolic concepts by interacting with their explanations. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3619–3629. IEEE, 2021.

Felipe Petroski Such, Aditya Rawal, Joel Lehman, Kenneth Stanley, and Jeffrey Clune. Generative teaching networks: Accelerating neural architecture search by learning to generate synthetic training data. In *International Conference on Machine Learning (ICML)*, pp. 9206–9216. PMLR, 2020.

Zhiqing Sun, Yikang Shen, Qinhong Zhou, Hongxin Zhang, Zhenfang Chen, David D. Cox, Yiming Yang, and Chuang Gan. Principle-driven self-alignment of language models from scratch with minimal human supervision. *CoRR*, 2023.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning (ICML)*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3319–3328. PMLR, 2017.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826. IEEE Computer Society, 2016.

Niket Tandon, Aman Madaan, Peter Clark, and Yiming Yang. Learning to repair: Repairing model output errors after deployment using a dynamic memory of feedback. In *Findings of the Association for Computational Linguistics (NAACL)*, pp. 339–352. Association for Computational Linguistics, 2022.

Stefano Teso and Kristian Kersting. Explanatory interactive machine learning. In *Conference on AI, Ethics, and Society (AIES)*, pp. 239–245. ACM, 2019.

Stefano Teso, Öznur Alkan, Wolfgang Stammer, and Elizabeth Daly. Leveraging explanations in interactive machine learning: An overview. *Frontiers in Artificial Intelligence*, 6, 2023.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

Lin Wang and Kuk-Jin Yoon. Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(6): 3048–3068, 2022.

Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M. Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3462–3471. IEEE Computer Society, 2017.

Xin Wang, Yudong Chen, and Wenwu Zhu. A survey on curriculum learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):4555–4576, 2022.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 13484–13508. Association for Computational Linguistics, 2023.

James Woodward. *Making things happen: A theory of causal explanation.* Oxford University Press, 2005.

Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific Data*, 10(1):41, 2023.

Matej Zečević, Devendra Singh Dhami, Constantin A Rothkopf, and Kristian Kersting. Causal explanations of structural causal models. *CoRR*, abs/2110.02395, 2021.

Eric Zelikman, Jesse Mu, Noah D Goodman, and Yuhuai Tony Wu. Star: Self-taught reasoner bootstrapping reasoning with reasoning. 2022.

Zijian Zhang, Koustav Rudra, and Avishek Anand. Explain and predict, and then predict again. In Liane Lewin-Eytan, David Carmel, Elad Yom-Tov, Eugene Agichtein, and Evgeniy Gabrilovich (eds.), *International Conference on Web Search and Data Mining (WSDM)*, pp. 418–426. ACM, 2021.

Appendix

In the following, you can find details on our LSX instantiations, experimental data and evaluation metrics.

A Model and Training Details

Exact implementation details can be found in the corresponding code.

A.1 CNN-LSX

The CNN-LSX configurations are identical for the MNIST and ChestMNIST dataset.

Learner.

The learner corresponds to a convolutional neural network with two convolutional layers, ReLU activation layers, one average pooling layer and two linear layers.

Critic.

The critic for CNN-LSX is identical to the architecture of the corresponding learner for both MNIST and ChestMNIST.

FIT.

Within the FIT module the learner is optimized via a cross-entropy loss as $l_{\mathrm{B}} := l_{\mathrm{CE}}^{f}(f(X_f), y_f)$.

EXPLAIN.

The explanation method of CNN-LSX corresponds to the InputXGradient method described in Shrikumar et al. (2017) and implemented via the captum[5] pytorch package. Following Ancona et al. (2018), for an input sample x_i and the output of model f given the corresponding ground truth label, y_i, it is defined as:

$$\mathrm{InpXGrad}_i = x_i \cdot \frac{\partial f_{y_i}(x_i)}{\partial x_i}.$$

REFLECT.

In the REFLECT module for CNN-LSX the critic trains for one epoch on the explanations obtained from the learner given the input data of $\bar{X}_c$, *i.e.*, E_c. Specifically, the critic here is trained via a cross-entropy loss to predict the corresponding class of the learner's explanations. We allow the critic to update its parameters while iterating over all batches in (E_c, y_c) whereby the loss values are accumulated over all batches and averaged. In practice we found that it was necessary to reinitialize the critic with each LSX iteration. The final accumulated and averaged loss value is passed back to the learner and represents the feedback `score` in CNN-LSX.

REVISE.

In a standard REVISE step the learner again performs the original base task via l_{B} while jointly optimizing for the feedback via the critic in the previous REFLECT step. Specifically, the learner optimizes a joint loss: $L = l_{\mathrm{B}} + \lambda l_{\mathrm{CE}}^{c}(c(E_c), y_c)$. Hereby, λ represents a scaling hyperparameter which we set quite high (*e.g.*, $\lambda \geq 100$) in our evaluations to prevent the learner from mainly optimizing for good prediction. Also here we refer to the corresponding code for the exact parameter values.

We perform the triad of LSX modules (EXPLAIN, REFLECT, REVISE) for several iterations until iteration budget T is reached.

As a final REVISE step, *i.e.*, when the iteration budget has been reached, we perform a fine-tuning step in which we let the learner produce explanations for all samples in X_f, E_f = EXPLAIN$(f, X_f | y_f)$, and let f be

---

[5] captum.ai/

optimized for the base task making sure that it does not diverge its explanations from E_f in the process. This is done via the combined loss $L = l_B + \lambda_{ft} l_{ft}(E'_f, E_f)$, where l_{ft} represents a simple mean-squared error loss between E_f and the explanations E'_f of each optimisation iteration.

A.2 NeSyCL-LSX

Learner.

The learner submodel for the NeSyCL-LSX instantiation differs for the CLEVR-Hans3 and CUB-10 datasets. For CLEVR-Hans3 the learner corresponds to the concept learner of Stammer et al. (2021) which incorporates a slot attention encoder for predicting the object's attributes and a set transformer for the final class prediction. As in Stammer et al. (2021), in our experimental evaluations, we make use of a pretrained perception (slot attention) module and perform updates only to the reasoning (set transformer) module, *i.e.*, the module making the final predictions. For the CUB-10 configuration the learner corresponds to the setup of Koh et al. (2020) representing an Inception-v3 model Szegedy et al. (2016) for predicting the bird concepts and a simple linear layer to make the final class prediction.

Critic.

The critic, c, of the NeSyCL-LSX instantiation, both for CLEVR-Hans3 and CUB-10, corresponds to the neural-symbolic forward reasoner of Shindo et al. (2021). Where the predicate specifications etc. required for the forward reasoner for CLEVR-Hans3 correspond to the original ones of Shindo et al. (2021) for CUB-10 we had to redefine each of the 28 bird concepts as neural predicates *e.g.*, `haswingcolor` can take six different values which in the notation of Shindo et al. (2021; 2023) is defined as: `haswingcolor:brown,grey,yellow,black,white,buff`. We refer here to our repository and the original work of Shindo et al. for details.

FIT.

Similar to the CNN-LSX instantiation the FIT module in NeSyCL-LSX corresponds to optimizing for class prediction via a cross-entropy loss $l_{\text{B}} = l_{\text{CE}(f(X_f),y_f)}$. As previously mentioned, in our evaluations we hereby freeze the parameters of the perception module of f, thus optimizing only the parameters of the reasoning (aka predictor) module of the learner.

EXPLAIN.

The EXPLAIN module of NeSyCL-LSX builds on the explanation approach of the concept learner of Stammer et al. (2021). Specifically, it first computes the integrated gradients Sundararajan et al. (2017) of the concept representation, z_i. Again following the notation of Ancona et al. (2018) this is defined as:

$$\text{IntGrad}_i = (x_i - \bar{x}_i) \cdot \int_{\alpha=0}^{1} \frac{\partial f_{y_i}(\tilde{x}_i)}{\partial \tilde{x}_i} |_{\tilde{x}_i = \bar{x}_i + \alpha(x_i - \bar{x})} d\alpha.$$

Hereby, $\bar{x}_i$ represents a "baseline" value, which in our evaluations corresponds to a zero vector. Next, the resulting importance map on the latent concept representations is binarized via a hyperparameter $\delta \in [0, 1]$ to $e''_i \in \{0, 1\}^{O \times A}$. We next propositionalize [6] e''_i by representing the explanation as a set of logical rules that are present in e''_i. These rules consist of conjunctive combinations of the important attributes and objects. We denote this set of candidate explanatory rules as e_i. For example, let us assume that for a specific CLEVR-Hans3 sample i of class 1 in the explanation form e''_i we identify two objects to be important for the final class prediction. Hereby, the attributes green color and spherical shape are important for the first object and cubical shape for the second object. Following the notation of Shindo et al. (2023) The set of possible conjunctive rules within this explanation, e''_i are now:

---

[6] Changing the representation of relational data.

```
class1(X):- in(O1,X),color(O1,green).
class1(X):- in(O1,X),shape(O1,sphere).
class1(X):- in(O1,X),shape(O1,cube).
class1(X):- in(O1,X),color(O1,green),shape(O1,sphere).
class1(X):- in(O1,X),in(O2,X),color(O1,green),shape(O2,cube).
class1(X):- in(O1,X),in(O2,X),shape(O1,sphere),shape(O2,cube).
class1(X):- in(O1,X),in(O2,X),color(O1,green),shape(O1,sphere),shape(O2,cube).
```

We refer to this step of constructing all potential candidate rules as propositionalizing.

Notably, each input sample thereby produces a set of such candidate rules which may potentially contain many duplicates over samples of the same underlying class. Finally, by iterating over all samples in X_c, grouping the resulting candidate rules by image class and removing duplicates we receive a set of candidate rules per class as $E_c = \{e^1, ..., e^K\}$.

For improved running times it is beneficial to limit the number of candidate rules per input sample by a maximum number of objects and attributes per object within an explanation rule *e.g.*, maximally four objects per rule. In our evaluations we set these two hyperparameters to still greatly overestimate the ground-truth rule and refer to the code for details (as well as for the values of δ). It is important to note that the propositionalizing step breaks the differentiality of the explanations.

Reflect.

Having obtained the set of candidate `explanation` rules per image class, we pass these candidate rules to the critic *i.e.*, forward reasoner of Shindo et al. (2021). For each underlying class and based on the data within X_c the critic next estimates the validity of each candidate rule, where we refer to Shindo et al. (2021) and Shindo et al. (2023) for details on this.

This evaluation is done for all positive examples of a class and for all negative examples of a class (*i.e.*, all remaining classes), resulting in two probabilities for the ith explanation rule e_i^j of class j. We denote these probabilities as σ_i^{j+} and σ_i^{j-}, respectively. The first probability represents the validity of the rule in samples of the relevant class (class j) and the second the validity in samples of all other (irrelevant) classes ($k \in \{1, ..., K\} \backslash j$).

As we consider an explanation to be good if it distinguishes an input sample from samples of opposite classes, but indicates similarities to samples of the same class, we next compute the probability for each candidate explanation rule as $\sigma_i^j = \sigma_i^{j+} - \sigma_i^{j-}$. These probabilities, σ, correspond to the `score` of NeSyCL-LSX and represent the numerical values in the `score` representation in Fig. 3. Finally, per image class, j, we select the explanation rule with the maximal probability from σ^j corresponding to the red enclosed rule in Fig. 3. We denote this as e^j_{max} for class j with $E_{\text{max}} = \{e^1_{\text{max}}, ..., e^K_{\text{max}}\}$ for the set over all classes. We thus pass E_{max} to the learner in the Revise module, rather than the raw `score` value.

Revise.

The selected explanations of the previous Reflect step, E_{max}, are next mapped back into binary vector form in the dimensions of the learner's symbolic representation latent concept representation $\bar{E}_{\text{max}} = \{\bar{e}^1_{\text{max}}, ..., \bar{e}^K_{\text{max}}\}$ with $\bar{e}^j_{\text{max}} \in \{0, 1\}^{O \times A}$. This is required so we can compare the learner's explanations to the valid explanations as identified by the critic in a differentiable manner. Specifically, in the Revise step of NeSyCL-LSX we optimize a joint loss function corresponding to $L = l^{\text{B}} + \lambda l_{\text{MSE}}(\bar{E}_{\text{max}}, E_f)$. For explanation e_i of input sample $x_i \in X_f$ with corresponding class label y_i l_{MSE} is defined as:

$$l_{\text{MSE}}(\bar{e}^{j}_{\text{max}}, e_i) = \frac{1}{O \times A} \sum_{q=1}^{O \times A} (e_{i_q} - \bar{e}^{y_i}_{\text{max}_q})^2.$$

In comparison to CNN-LSX in this instantiation $T = 1$. Meaning the critic only scores the proposed underlying logical explanations once and passes this back as feedback. Although it is in principle possible to perform multiple steps of this, *e.g.*, by first removing explanations which are most unprobable from the learner's representations and only after several of such iterations choose the most likely explanation, we leave this for future investigations.

B Data

CUB-10. CUB-10 represents a subset of the original Caltech-UCSD Birds-200-2011 dataset (CUB-200-2011) (Wah et al., 2011) that comprises images of the first 10 classes of the full dataset. Koh et al. (2020) originally perform a preprocessing step for CUB-200-2011 where concept vectors are replaced with the max voting vector over all samples of a class. In other words, the resulting concept activations are identical across all samples of a class which leads to a one-to-one mapping between concept activations and the class affiliation.

In CUB-10 we simulate a more realistic setting in which the class concept activations of (Koh et al., 2020) are overlaid with additional random noise, thereby maintaining the underlying class-based concept activation, but producing random variations per class sample. Specifically, we add uniformly distributed noise between 0 and 1 onto the class-based concept activations and binarize the resulting activations with a threshold of 0.75.

C Metrics

C.1 Explanation Consolidation.

Ridge Regression Classification. For evaluating the separability of learned explanations, we provide the accuracy of a ridge regression (RR) model that is fitted on a set of tuples consisting of explanations from a trained learner and the corresponding ground-truth (GT) class labels of the underlying image. The RR model is fitted on a training set and tested on an additional, held-out set of explanations (and corresponding class labels).

This evaluation acts as a proxy of the separability of learned explanations. The higher the RR accuracy on the test set the better the separability between explanations. For each learning configuration in our evaluations we train a RR model separately on the explanations from the five differently seeded models.

Encoding Analysis. The Inter- vs Intraclass Explanation Similarity (IIES) is defined as:

$$\text{IICS} = \frac{1}{K} \sum_{k}^{K} \frac{\frac{1}{M} \sum_{i}^{M} d(z^k_i, \mu^k)}{\frac{1}{K} \sum_{j, j \neq k}^{K} d(\mu^j, \mu^k)}$$

Essentially, this metric estimates in how far the explanations stemming from samples of one class are close to another compared to the explanations of samples from other classes. The encoding of a pretrained model, h, provides the encoding space in which this similarity is assessed. The lower the values of IICS the better separable are the data for h.

Here z^k_i corresponds to the encoding of the explanation of a sample i from class k. This encoding is provided by an additional model, h, via $h(e_i) = z^k_i$, where e_i is a provided explanation of sample i from a learner f. h is identical in architecture to the learner of which the explanations are being evaluated, however h was separately pretrained only on the original task. Specifically, for evaluating explanations from the CNN configurations, h corresponds to an identical CNN that was trained for image classification as in the vanilla configurations. For evaluating the NeSyCL configurations a NeSyCL was pretrained for classification as in the NeSyCL vanilla setting. In both cases h was provided with a random seed different from those used in the original training setups.

Furthermore, μ^k corresponds to the average encoding over all samples of class k (where for notations sake we assume M samples in each class, although this can vary in practice). $d(x, y)$ represents a distance metric between x and y, where we have used the euclidean distance in our evaluations. We divide the distance within one class by the average distance between the encoding mean of class k and those of all other classes, corresponding to an estimate of the distance to all other class encodings. Finally this is averaged over all classes.

C.2 Faithfulness

For comprehensiveness, parts of the input are removed that correspond to important features as identified by the explanation. As a result, the model should be less accurate in its predictions. In the case of sufficiency, one removes those input features which were deemed unimportant according to the explanation. Hereby, the model should not lose much accuracy. Notably, the original sufficiency and comprehensiveness metrics of (DeYoung et al., 2020) were introduced in the context of NLP in which input sequences are considered as discrete inputs. However, removing input features from continuous inputs such as images presents an issue (Hooker et al., 2019) as measured differences due to pixel removal may reflect the influence of the modified, out-of-distribution input rather than faithfulness of the explanation. For this case, we modified the metrics for the CNN configurations (*i.e.*, for explanations that are in a continuous form) to approximately compensate for this effect. For evaluating explanation faithfulness we thus provide results for CNN-LSX (and vanilla CNN) via the continuous adaptation of both metrics (denoted as $\text{COMP}_{cont.}$ and $\text{SUFF}_{cont.}$) and for NeSyCL-LSX (and NeSyCL vanilla) via the original comprehensiveness and sufficiency definitions (denoted as $\text{COMP}_{discr.}$ and $\text{SUFF}_{discr.}$). We formalize these in the following.

We follow the notation for $\text{COMP}_{discr.}$ and $\text{SUFF}_{discr.}$ of Chan et al. (2022). For this, x denotes an input sample. We denote the predicted class of x as $c(x)$, and the predicted probability corresponding to class j as $p_j(x)$. Assuming an explanation is given, we denote denote the input containing only the $q\%$ important elements as $x_{:q\%}$. We denote the modified input sequence from which a token sub-sequence x' are removed as $x \setminus x'$. Comprehensiveness and sufficiency for discrete explanations are finally defined as:

$$\text{COMP}_{\text{discr.}} = \frac{1}{|B|} \sum_{q \in B} \frac{1}{N} \sum_{j=1}^{N} (p_{c(x_j)}(x_j) - p_{c(x_j)}(x_j \backslash x_{j_{:q\%}}))$$

$$\text{SUFF}_{\text{disc.}} = \frac{1}{|B|} \sum_{q \in B} \frac{1}{N} \sum_{j=1}^{N} (p_{c(x_j)}(x_j) - p_{c(x_j)}(x_{j_{:q\%}})).$$

Where N here represents the number of data samples in the evaluation set. In our evaluations we set $B = \{1, 5, 10, 20, 50\}$ as in the original work of DeYoung et al. (2020).

For computing comprehensiveness and sufficiency scores based on continuous explanations we first compute the comprehensiveness and sufficiency when a percentage q of the top input elements (*e.g.*, pixels) are set to the median value of all input elements of the evaluation set. In comparison to the definition of $\text{COMP}_{discr.}$ and $\text{SUFF}_{discr.}$ of DeYoung et al. (2020) for the adaptation to continuous explanations we base the metrics on class accuracy rather than class probabilities. We denote these alternative computations as:

$$\hat{\text{COMP}}_{\text{cont.}} = \frac{1}{B} \sum_{q \in B} \text{acc}(f(X \backslash X^{\text{median}}_{:q\%}), y)$$

$$\hat{\text{SUFF}}_{\text{cont.}} = \frac{1}{B}\sum_{q\in B} \text{acc}(X_{:q\%}^{\text{median}}, y).$$

Here, $\text{acc}(f(X), y)$ corresponds to the accuracy score of a models prediction given input data, $f(X)$, compared to the ground truth labels, y. $X_{:q\%}$ corresponds to the full dataset in which everything but the top $q\%$ of each samples input elements were set to the median value of the dataset and $X\backslash X_{:q\%}^{\text{median}}$ where the top $q\%$ of each samples input elements were set to the median value of the dataset.

Next we compute the same metrics, but when removing randomly chosen $q\%$ of the input elements by setting them to the median value. We denote these computations as $\hat{\text{COMP}}_{cont.}^{rand}$ and $\hat{\text{SUFF}}_{cont.}^{rand}$. Finally, we subtract these from the original values, leading to:

$$\text{COMP}_{\text{cont.}} = \hat{\text{COMP}}_{cont.}^{rand} - \hat{\text{COMP}}_{cont.}$$

and

$$\text{SUFF}_{\text{cont.}} = \hat{\text{COMP}}_{cont.}^{rand} - \hat{\text{COMP}}_{cont.}$$

C.3 Self-unconfounding

For the *w/ deconf.* case we provide the critic in each case with samples from the original test sets, whereby we remove these samples from the final test set evaluations. In the baseline configurations we add these samples to the learner's training dataset for a fair comparison. Specifically, for evaluations on CLEVR-Hans3 *w/ deconf.* we provide the critic with 150 randomly selected sampled from the original test set. For DecoyMNIST we provide the critic with 1024 test set samples.