Algebraic Machine Learning: learning as computing subsets of the subdirect decomposition from Abstract Algebra

Fernando Martin-Maroto
Champalimaud Foundation, Lisbon, Portugal
fernando.martinmaroto@neuro.fchampalimaud.org

Nabil Abderrahaman-Elena
Algebraic AI, Madrid, Spain
nabil.abderrahaman@gmail.com

David Méndez
Champalimaud Foundation, Lisbon, Portugal
david.mendez@research.fchampalimaud.org

Gonzalo G. de Polavieja
Champalimaud Foundation, Lisbon, Portugal
gonzalo.polavieja@neuro.fchampalimaud.org

Abstract

We propose a machine-learning framework based on the principle that learning can be achieved through algebraic decomposition rather than numerical optimization. A learning task is encoded as axioms of an algebra, and learning proceeds by computing a subdirect decomposition of the algebra into subdirectly irreducible components. We show that suitable subsets of these components form models that generalize to unseen data because they approximate the underlying rule in the training data. The method has no dataset-specific hyperparameters to tune and shows no evidence of classical overfitting in our experiments, so we run it without a validation dataset. We compare against multilayer perceptrons as generic parametric baselines without task-specific inductive biases. Across 24 classification datasets, the algebraic method is applied only once to each training set without hyperparameter tuning, whereas the multi-layer perceptrons are selected from many configurations using validation data or cross-validation. Nevertheless, the resulting test accuracies are statistically indistinguishable. Moreover, adding a statistical readout on top of the algebraic model, using untuned logistic regression, improves over the best multilayer perceptrons selected by validation or cross-validation. More broadly, because task axioms may encode examples, formal knowledge, or both, this approach provides a common algebraic framework and the same algorithm for data-driven problems such as classification, formally specified problems such as computing Hamiltonian cycles from specifications, and mixed settings combining data with formal constraints.

1 Introduction

A common strategy for analyzing complex mathematical structures is to decompose them into simpler components. For example, Fourier analysis decomposes signals into frequency components [1, 2] and spectral theory decomposes matrices into eigenspaces or spectral components [3, 4]. These decompositions are useful because they expose components that can be selected, discarded, or recombined to capture the structure relevant to a task.

Here we ask whether an analogous decomposition principle can be used for learning from data. Our starting point is Birkhoff's subdirect representation theorem: for any algebra A, there exist algebras $B_1, \ldots, B_n$ and an injective homomorphism

$$\eta : A \to B_1 \times B_2 \times \cdots \times B_n, \tag{1}$$

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

where the factors B_i can be chosen to be subdirectly irreducible [5, 6]. In this paper, we turn this existence theorem into a learning method. A learning task is encoded in the algebra A, a subdirect representation is computed, and learning consists of selecting those components B_i that capture the rule underlying the data and therefore generalize to unseen examples. The specific contributions of this paper are:

1. We formulate machine learning as component selection in a subdirect decomposition, and show how to compute the relevant components in the semilattice setting.

2. We prove that, although subdirect decompositions are non-unique, the freest model recovers the hidden rule underlying the data given sufficient data.

3. For finite training sets, we provide a probabilistic analysis characterizing which components of the subdirect decomposition are expected to achieve low test error.

4. We develop a theoretical formulation of Sparse Crossing, the algorithm to obtain these low test error components of the subdirect decomposition, and extend the initial algorithm introduced in [7].

5. Across 24 classification datasets, we evaluate the algebraic method without hyperparameter tuning or validation-based model selection. Despite this, its test accuracies are statistically indistinguishable from multilayer perceptrons (MLPs) selected over many hyperparameter configurations using validation data or cross-validation. Moreover, a logistic-regression readout on the algebraic model is significantly better than the best validation- or cross-validation-selected MLPs.

2 Related work

Algebraic Machine Learning. We introduced Algebraic Machine Learning (AML) in [7], including its algorithms Full Crossing and an initial version of the Sparse Crossing algorithm evaluated on binarized two-class MNIST [8]. Then we built the mathematics of atomized semilattices and its connection to the subdirect product [9, 10]. However, this earlier work did not show how the mathematics of [9, 10] could be used to understand and extend AML. Here, we establish that connection by interpreting learning as the selection of components in a subdirect decomposition. This gives a formal account of what AML learns and why it generalizes, and gives a new Sparse Crossing algorithm with a theory of how it identifies generalizing components. This extends AML to multiclass problems with numerical and categorical variables.

Symbolic and neurosymbolic AI. Classical symbolic AI represents knowledge through designer-specified rules, logical relations, or algebraic structures [11, 12, 13, 14]. Neurosymbolic systems combine such structure with neural models, using symbolic knowledge to constrain, guide, or regularize learning [15, 16, 17, 18]. In contrast, using algebraic decomposition we can infer rules from data and build predictors rather than regularizing a neural model with symbolic constraints.

Rule, concept, and lattice-based learning. Rule and concept learning, including inductive logic programming and formal concept analysis, recover structured rules or concepts from data [19, 20]. Formal concept analysis organizes data through concept lattices, closure systems, and Galois connections [20, 21, 22]. Our use of atomized semilattices differs in mechanism: we build a subdirect representation of the learning task and select components of that representation as the predictive model, rather than searching over logical rules, clauses, or concept lattices.

Algebraic structure in machine learning. Algebraic ideas appear in symmetry-aware neural architectures [23, 24], algebraic statistics [25], and category-theoretic frameworks for learning and compositionality [26, 27]. These works use algebra to impose structure, study statistical models, or organize compositional systems. We use the subdirect decomposition to construct the predictive model itself.

3 Framework

3.1 Encoding a machine learning task as axioms of a semilattice

An algebra in the sense of universal algebra is a non-empty set S together with one or more operations [6, 28, 29]. We work with a *semilattice*, that is, an algebra with the set S and a binary operation $\odot$ that satisfies the axioms of commutativity, associativity, and idempotency [6, 30].

We distinguish a set of special elements $C \subset S$, called *constants*, which serve as primitives for describing a machine learning task. In image classification, constants may for example represent pixels in a color, or class labels. Besides the constants, the set S contains all *terms* generated from the constants under repeated application of the semilattice operation $\odot$. Thus, if the set of constants is $C = \{c_1, \ldots, c_m\}$, then an expression such as $T = c_2 \odot c_5 \odot c_9$ is a term, in this case with component constants $C(T) = \{c_2, c_5, c_9\}$.

We encode the training data of a machine learning task by adding additional axioms to the semilattice. Each such axiom asserts a relationship between two terms, say T_L and T_R, of the positive form as $T_R \odot T_L = T_R$ or of negative form $T_R \odot T_L \neq T_R$, which we write for short using the standard semilattice order notation as $T_L \leq T_R$ and as $T_L \not\leq T_R$, respectively. We read $T_L \leq T_R$ as saying that T_L is contained in, or implied by, T_R. For example, in a classification task we can express that an example is of a given class i as the task axiom $T_L \leq T_R$ with $T_L = class_i$ and $class_i$ the constant encoding the label of class i, and T_R encoding the example, say a 50×50 black and white bi-level image as $T_R = c_{11w} \odot ... \odot c_{50\,50\,b}$ with the constants c_{ijw} (c_{ijb}) representing the white (black) pixel in matrix position (i, j).

3.2 Computing the subdirect decomposition using an atomized semilattice

The next step is to compute a subdirect representation of the semilattice we have used to encode a task. We do this using an *atomized semilattice* [9], an enrichment of the semilattice that introduces additional elements, called *atoms*. Theorem 18 in Appendix A demonstrates that each atom of the atomized semilattice is in bijective correspondence to a subdirectly irreducible component in the subdirect representation of the semilattice encoding the task. In this and the next sections we show how to use atomized semilattices to learn a model.

An atomized semilattice consists of two disjoint sets, S and A, together with two binary operation symbols, $\odot$ and $<$. $(S, \odot)$ is the semilattice we have considered so far. The set A contains the atoms, which we denote by Greek letters. We interpret $<$ as a binary *relation* on $S \cup A$ that is reflexive and transitive.

The axioms defining how $<$ interacts with the elements of $S \cup A$ are given in Appendix A, Definition 7. Here we state briefly two of these axioms. One axiom says how atoms and constants are related: For each atom $\phi \in A$, there exists at least one constant $c \in S$ such that $\phi < c$. A second axiom says how atoms and terms are related: Every term $T \in S$ has at least one atom $\phi \in A$ such that $\phi < T$. The way we read $\phi < c$ (or $\phi < T$) is by saying that "c (or T) is in the *upper segment* of ϕ" or, equivalently, that "ϕ is in the *lower segment* of c (or T)."

An atomized semilattice model is an atomized semilattice that satisfies the task axioms. To know how the task axioms translate into conditions on atoms, we use the following consequence of the axioms of the atomized semilattice: $T_L \leq T_R$ is positive in the model if and only if every atom in the lower segment of T_L is also in the lower segment of T_R (Appendix A, Theorem 1(vi)):

$$T_L \leq T_R \iff \{\phi \mid \phi < T_L\} \subseteq \{\phi \mid \phi < T_R\}. \tag{2}$$

Otherwise $T_L \not\leq T_R$. To apply Eq. (2) in practice, we need to know how to compute the set of atoms in the lower segment of a term T, namely $\{\phi \mid \phi < T\}$. For this, we use another consequence of the axioms: the lower segment of a term T, with $T = c_1 \odot c_2 \odot \cdots \odot c_n$, is the union of the lower segments of its component constants (Appendix A, Theorem 1(v)):

$$\{\phi \mid \phi < T\} = \cup_{i=1}^{n} \{\phi \mid \phi < c_i\}. \tag{3}$$

Equations (2) and (3) show that training an atomized model amounts to synthesizing atoms so that the subset condition in Eq. (2) holds for all positive task axioms and fails for all negative task axioms. Equations (2) and (3) are also the expressions we use to determine whether a test example is positive or negative.

3.3 Some properties of the atomized semilattice

We use some properties to give appropriate notation to atoms and models. An atom ϕ is fully characterized by the constants in its upper segment (Appendix A, Theorem 1 (iv)), so we denote an atom by listing its upper-segment constants in square brackets. For example, $\phi[c_3, c_4]$ denotes the atom whose upper segment is exactly $\{c_3, c_4\}$, that is, $\phi < c_3$ and $\phi < c_4$ and ϕ is not in the lower segment of any other constant.

We can query whether any expression of the form $T_L \leq T_R$ is positive or negative in a model using the atoms and Equations (2) and (3). Therefore, a model M is completely characterized by its set of atoms. For example, we may write a model as:

$$M \;=\; \big\{\phi[c_1, c_2, c_3],\ \phi[c_2, c_5],\ \phi[c_1, c_6],\ \phi[c_3],\ \phi[c_3, c_4]\big\}. \tag{4}$$

Also, we say that an atom is redundant when its upper segment can be expressed as a union of upper segments of other atoms in the model. A redundant atom can be removed from the model without affecting whether any expression is positive or negative in the model (Appendix A, Theorem 5).

3.4 Freest atomized model

There are many possible atomized models of the same training task axioms differing in their behavior for test examples. We focus on the *freest atomized model*. The freest model is the model in which the positive expressions are only (a) the positive task axioms and (b) the positive logical consequences of the positive task axioms. All remaining expressions are negative. By positive logical consequences we mean those expressions that are positive in every model of the task axioms.

We provide *Full Crossing* as a constructive algorithm to compute the freest atomized model of a set of task axioms (Appendix A, Theorem 11). Concretely, suppose X is the set of positive task axioms already satisfied by a model M, and we wish to make positive a new task duple $T_L \leq T_R$ that is currently negative in the model M. Full Crossing produces the freest atomized model with positive axioms $X \cup \{\, T_L \leq T_R\}$.

Full Crossing works as follows. Let R be the set of atoms in the lower segment of T_R,

$$R(T_L \leq T_R) := \{\phi \mid \phi < T_R\}, \tag{5}$$

of a cardinality $n = |R|$. Let the discriminant D be the set of atoms in the lower segment of T_L that are not in the lower segment of T_R,

$$D(T_L \leq T_R) := \{\phi | \phi < T_L\} \setminus R(T_L \leq T_R), \tag{6}$$

If $D(T_L \leq T_R)$ is non-empty, then we have that $T_L \not\leq T_R$ in the model because the subset condition in Eq. (2) fails. Full Crossing eliminates this failure by replacing each atom in the discriminant, $\phi \in D$, by n new atoms. Each of these n new atoms has as its upper segment the union of (i) the upper segment of ϕ and (ii) the upper segment of one atom in R.

Example 1. Consider the expression $T_L \leq T_R$ with $T_L = c_1 \odot c_2$ and $T_R = c_3 \odot c_4$, which is negative in the model M in Eq. (4). We obtain $R = \{\phi[c_1, c_2, c_3],\ \phi[c_3],\ \phi[c_3, c_4]\}$ and $D = \{\phi[c_2, c_5],\ \phi[c_1, c_6]\}$. We form the following table with the atoms in D arranged in a column on the left and the atoms in R as a row at the top. The new atoms, shown on a gray background in the table, have the constants in the square brackets as the union of the constants of the atom in R in the same column and the constants of the atom in D in the same row:

D \ R	$\phi[c_1, c_2, c_3]$	$\phi[c_3]$	$\phi[c_3, c_4]$
$\phi[c_2, c_5]$	$\phi[c_1, c_2, c_3, c_5]$	$\phi[c_2, c_3, c_5]$	$\phi[c_2, c_3, c_4, c_5]$
$\phi[c_1, c_6]$	$\phi[c_1, c_2, c_3, c_6]$	$\phi[c_1, c_3, c_6]$	$\phi[c_1, c_3, c_4, c_6]$

We then replace the two atoms in the discriminant D with the six new atoms from the table, giving the new model satisfying $T_L \leq T_R$. The atoms $\phi[c_2, c_3, c_4, c_5]$ and $\phi[c_1, c_3, c_4, c_6]$ are redundant in the new model and can be removed.

To compute the freest model for a set of positive task axioms, we apply Full Crossing sequentially to each positive task axiom, with ordering irrelevant (Appendix A, Theorem 12). We also need an initial freest model, which has one atom per constant, (Appendix A, Theorem 13)

$$M_0 = \{\phi[c_1],\ \phi[c_2],\ \ldots,\ \phi[c_m]\}, \tag{7}$$

with $m = |C|$ the number of constants.

Example 2. Consider the task of assigning a property p to the set of 3,375 bi-level 4×4 images whose first column is entirely black, while the remaining three columns may contain black and white pixels but never form an additional fully black column (Figure 1a shows 16 examples). For this task,

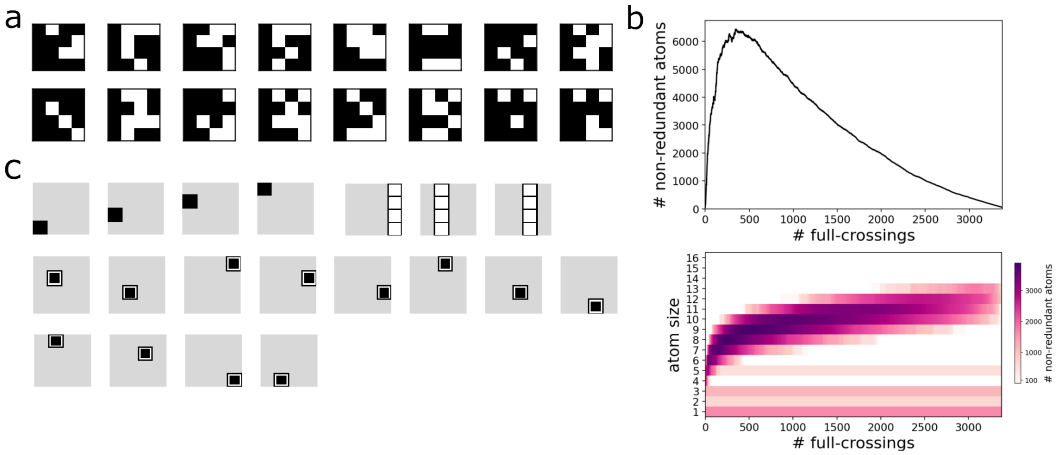


Figure 1: **Freest models for a toy task using images with only the first column in black. (a)** Sixteen of the 3,375 bi-level 4×4 images whose only fully black column is the first. **(b)** Top: Number of non-redundant atoms after a number of full-crossings of the task axioms. Bottom: Darker color indicates higher number of non-redundant atoms of a given size (with size being number of constants in their upper segment) and given a number of full-crossings. **(c)** Atoms of the final model.

we introduce 32 constants for the 16 pixel locations, with one constant for "white" and one for "black" at each location. We also introduce a constant p for the property, giving a total of 33 constants. Our initial model is one atom per constant, as in Eq. (7).

We apply the Full Crossing procedure sequentially to all 3,375 task axioms of the form $p \leq T_i$, where each T_i is a term comprising 16 component constants that represent one of the 4×4 images. As Full Crossing enforces more task axioms, the number of non-redundant atoms first increases to roughly 6,000 and then decreases to a final number of 51 (Figure 1b, top). Of these 51, 32 are atoms of the initial model, and the 19 remaining ones correspond to the rule underlying the data (Figure 1c): four atoms capture that every positive image contains a black pixel in rows $i = 1, 2, 3, 4$ of the first column, that is, a black vertical bar as the first column, 3 atoms capture that positive images have columns 2, 3 or 4 with at least a pixel in white, and 12 atoms that capture that positive images can have each pixel in the last three columns as either black or white.

3.5 Freest atomized model gives the underlying rule in the data

In the previous section, we found that the model converged to the rule underlying the data. This phenomenon is general, as formalized by the following theorem (Appendix B, Theorem 23):

Theorem. *Let P and Q be two sets of positive task axioms such that Q is the set of logical consequences of P minus the set P. The non-redundant atoms of the freest atomized model of Q consist of non-redundant atoms of the freest atomized model of P and an additional set of atoms of cardinality equal or lower than $|P|$.*

In a learning setting, the training is a subset of Q, that is, some of the logical consequences of the hidden rule axioms P. According to the theorem, the freest model trained on enough of Q approaches the freest model of the hidden rule P, except for a small number of atoms of cardinality at most $|P|$ because the training examples do not contain the hidden rule P as explicit axioms.

The atoms of the hidden rule P can already emerge in the practical cases of very small subsets of Q. In the toy example of the previous section, for example, all the non-redundant atoms of the hidden rule are already present after 200 full-crossings (Figure 1c shows convergence of the number of atoms at atom sizes of 2, 3 and 5 already at approximately 200 full-crossings). Extracting these non-redundant

atoms after 200 full-crossings, or even approximations to the atoms before 200 full-crossings, would provide a generalizing model, as we discuss in the next section.

3.6 Subsets of the freest atomized model are models that generalize

As we saw in Section 3.4, one of two things happens to each atom at each Full Crossing step. If the atom is not one of the atoms that makes negative the training axiom we want to make positive, it remains unchanged. Otherwise, it gets replaced by one or more atoms with an upper segment containing additional constants (for a more formal account, see Appendix A, Theorem 22). Because there are only finitely many constants, an atom can grow only finitely many times until it reaches the form associated with the hidden rule, and then cannot change anymore. This is the algebraic picture of the emergence of atoms of the hidden rule before all examples of the rule have been seen.

The following probabilistic analysis shows that atoms need not be of the matured hidden rule atoms to be useful for a model that generalizes. Suppose we select a subset of Z atoms from the freest atomized model at step n in training, and denote its atoms as $\phi_1, \phi_2, \ldots, \phi_Z$. Each atom ϕ_i has remained unchanged $h(\phi_i)$ times since its creation in training. The probability of a false negative on the test set, PFN, is the probability that at least one of these Z atoms causes a test example that should be positive to be negative in the model. Assuming that atom-level false-negative events are approximately independent and that the training and test distributions are the same, the expected test PFN for the atom subset can be approximated by (Appendix C):

$$\overline{\mathrm{PFN}}(\phi_1, \ldots, \phi_Z) \approx \sum_{i=1}^{Z} 1/h(\phi_i). \tag{8}$$

This expression shows that false negatives are reduced by selecting fewer atoms and by preferring atoms with larger success counts $h(\phi_i)$.

The probability of a false positive on the test set, PFP, is the probability that every one of the Z atoms fails to discriminate the test example. Assuming that the atom-level false-positive events are independent, the subset PFP factorizes as:

$$\overline{\mathrm{PFP}}(\phi_1, \ldots, \phi_Z) = \prod_{i=1}^{Z} \mathrm{PFP}(\phi_i). \tag{9}$$

where $\mathrm{PFP}(\phi_i)$ can be estimated empirically from the negative training task axioms under the standard assumption that the training and test distributions match. Equation (9) then implies that false positives are reduced by selecting enough atoms that are individually effective at discriminating negative training examples.

Eqs. (8) and (9) assume independence of atom-level false negatives and positives. When atom-level false-negative events are positively correlated, Eq. (8) is conservative and the $\overline{\mathrm{PFN}}$ may be smaller. When failures of atoms to discriminate negative examples are positively correlated, Eq. (9) is optimistic. Consequently, good subsets should not only contain atoms with low individual PFN and PFP, but also with low correlation in their discrimination behavior on negative training axioms.

We have thus obtained a principled method for extracting a generalizing model from the freest atomized model. To reduce test $\overline{\mathrm{PFN}}$, one should select as few atoms as possible that have been stable as possible (high $h(\phi_i)$). To reduce test $\overline{\mathrm{PFP}}$, the selected atoms should be individually effective at discriminating negative training axioms, exhibit low mutual correlation in their discrimination behavior, and be sufficiently numerous to ensure that every negative duple in the training set remains negative.

3.7 Sparse crossing as a practical computation of generalizing models

In practice, the freest model of a realistic task is typically far too large to compute explicitly. Since our goal is not the full freest model but a generalizing subset of its atoms, we use *Sparse Crossing*, a sparse variant of Full Crossing that directly seeks such subsets. Details of the Sparse-Crossing algorithm, including pseudocode, are provided in Appendix D. Open-source code and three notebooks are also available (Appendix E.2).

The main idea of Sparse Crossing is to select from the Full Crossing table, for example the one shown in Section 3.4, a subset of the atoms from the table that discriminate the negative task axioms, and

discard the rest. However, this simple procedure does not work because an atom that is safe to discard at one crossing step may later be needed to keep some negative task axiom negative after further crossings. Instead, we use an invariance property that allows atoms to be discarded while guaranteeing that negative task axioms remain negative after all positive axioms have been enforced (Appendices D.4 and D.5). This procedure is used in batches, and we additionally include a mechanism to improve cross-batch consistency by adding auxiliary negative axioms derived from atoms obtained in previous batches, which also leads to faster convergence (Appendix D.8 and Theorem 38).

In each batch, Sparse Crossing produces a new model that we call *master model*, as well as a model that aggregates this master model with the atoms of previous master models, the *union model*. After a sufficient number of batches, the union model satisfies all training axioms and we stop training. No validation dataset is required for this stopping criterion. For practical purposes, we use a model smaller than the final union model. Concretely, we extract multiple subsets of atoms from the final union model that satisfy all negative task axioms, and we take the union of these subsets until the model size reaches approximately 10% of the original union model (Appendix E.6). We refer to the resulting model as *AML model*.

4 Experiments

4.1 Simple encoding of a classification task

The simple encoding introduced in Example 2 of Section 3.4 can be used for any bi-level image classification task. We used the MNIST dataset [8] as a natural testbed for this encoding because the original digit images were bi-level images, with grayscale values arising only from size-normalization and centering into a 28×28 frame. For our encoding, we binarized MNIST by thresholding the image at a pixel value of 25. We used $2 \times 28 \times 28$ constants to represent the two possible states (black or white) of each pixel location, and ten class constants $digit_i$ for $i \in \{0, \ldots, 9\}$. Each of the $50{,}000$ MNIST training examples was encoded as a positive task axiom of the form $digit_i \leq image_{k,i}$, where $image_{k,i}$ denotes the term encoding the k-th training image of class i with $28 \times 28 = 784$ component constants. We also included the $9 \times 50{,}000 = 450{,}000$ negative task axioms $digit_j \not\leq image_{k,i}$ with $j \neq i$.

Using this encoding, we applied Sparse Crossing until the $500{,}000$ training task axioms were satisfied by the union model (Figure 2a, red dot). Some of the resulting atoms can be visualized in Appendix E.12.

We then extracted an AML model of approximately 10% of the size of the union model. The AML model did not empirically show overfitting, that is, its test accuracy did not drop, even when the union model is at 100% accuracy (Figure 2a, black points after the vertical dashed line). Since we observed no overfitting and did not tune task-dependent hyperparameters, we did not use a validation dataset for any AML experiment in this paper.

Using the AML model, a test image was classified as digit i if $digit_i \leq image_{test}$ or, equivalently, if:

$$D(digit_i \leq image_{test}) = 0, \tag{10}$$

with D the discriminant in Eq. 6. In our evaluation, $4{,}878$ of the $10{,}000$ test images satisfied this condition for exactly one digit class. These $4{,}878$ images were classified with 100% accuracy, indicating that the learned model captured a rule that perfectly covered roughly half of the test set.

To assign labels to all test images, we used a simple best-guess rule based on how close each class is to satisfying $D(digit_i \leq image) = 0$. For a test image $image_{test}$, we chose the label $i^\star$ corresponding to the smallest discriminant:

$$i^\star(image) = \underset{i \in \{0,1,\ldots,9\}}{\arg\min} \left| D(digit_i \leq image_{test}) \right|. \tag{11}$$

We refer to the reduced model together with this best-guess discriminant assignment rule as the AML model result, which gives a test accuracy of 97.63% (Figure 2b, black vertical bar). As an alternative to the best-guess discriminant assignment rule, we used logistic regression as a simple way to incorporate statistical weighting on top of the AML model (Appendix E.7). We trained the weights until the training accuracy reached 100%, and we again did not use any validation dataset. This gave a test accuracy of 98.26% (Figure 2b, gray vertical bar).

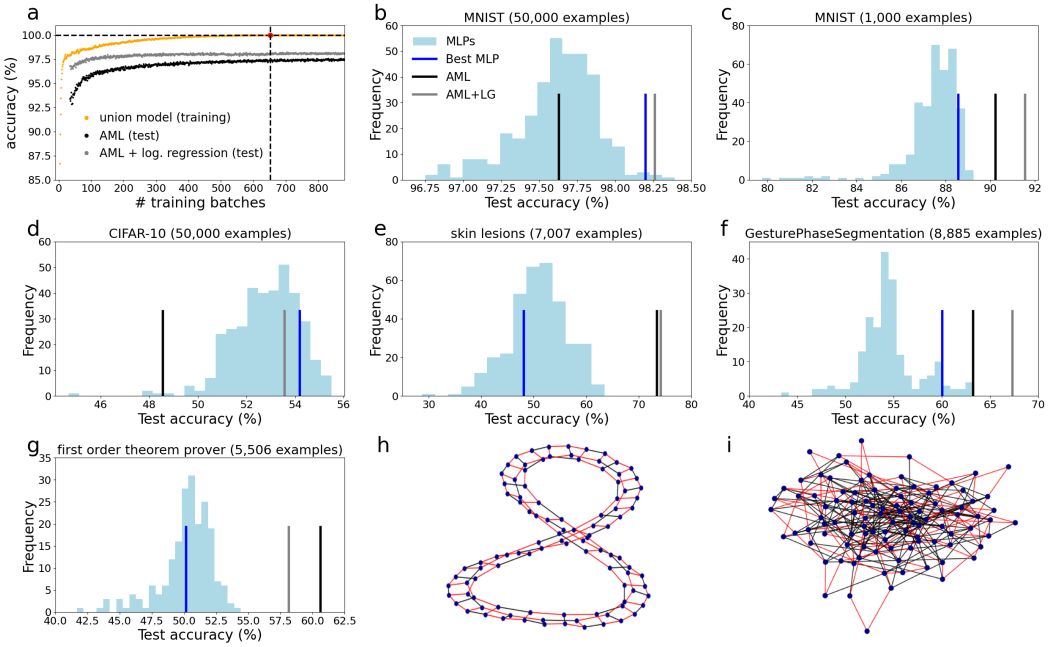


Figure 2: **Example experiments in AML. (a)** Training accuracy of the union model on the MNIST dataset [8], achieving 100% at the red dot. Test accuracy of the AML model result (black) and of the AML model followed by logistic regression using only training data (gray). **(b)** Test-accuracy distribution on MNIST for 360 MLPs trained with different hyperparameter configurations (cyan). Vertical bars show the test accuracy of the best validation-selected MLP (blue), the AML model result (black), and the AML model followed by logistic regression trained only on the training set (gray). **(c)** Same as (a) but for 1000 training examples. **(d,e)** Same as (a) for CIFAR-10 [31] and skin lesions [32], respectively. **(f,g)**. Same as (a) for tabular datasets from [33] and [34]. **(h,i)** Hamiltonian cycles (red) found by AML for the G7 graph from the FHCP challenge set [35] and for a random graph.

Our AML encoding did not incorporate image-specific inductive biases, so we compared against multilayer perceptrons (MLPs), which also lack such biases. We trained 360 MLPs over a broad hyperparameter grid, with two, three or four hidden layers (Appendix E.8). Their test accuracies had mean 97.61% and standard deviation 0.27% (Figure 2b, cyan). The validation-selected MLP, with layer widths $4096, 256, 256, 10$, reached 98.20% test accuracy (blue line), slightly below AML followed by logistic regression at 98.26%. We also evaluated a low-data setting with $1{,}000$ training examples, while still allowing MLP model selection on $10{,}000$ validation examples. AML and AML followed by logistic regression achieved 90.24% and 91.56% test accuracy, respectively (black and gray lines in Figure 2b), exceeding all trained MLPs with 360 different hyperparameter configurations.

We studied some properties of the AML models. First, we studied the variability in accuracy across different training runs. Using 10 training runs, we obtained that AML using the best-guess discriminant rule achieved an accuracy of 97.39% (mean) $\pm$ 0.15% (standard deviation), and with logistic regression on top 98.19% (mean) $\pm$ 0.07% (standard deviation). In contrast, the MLP pipeline shows a higher standard deviation, with 97.94% (mean) $\pm$ 0.69% (standard deviation). See details of runs in Appendix E.10. We also empirically studied the behaviour of AML with respect to noise, in the form of random pixel inversion as well as random label flips, and found a behavior comparable to MLPs, Appendix E.9.

4.2 Using formal knowledge in the task axioms

Task axioms can also encode purely formal knowledge, such as the specification of a Hamiltonian cycle. In this case, one axiom we use is the following. For each node i in a graph, and for each couple of edges E_y and E_z, we use the task axiom

$$P \odot \left(\odot_{x:\, x \notin \{y,z\}} nE_x \right) = E_y \odot E_z \odot P, \tag{12}$$

where nE_i denotes a constant representing the absence of edge E_i. This axiom states that, within the path P, selecting the two incident edges E_y and E_z is equivalent to excluding every other edge incident to node i. Together with the remaining axioms in Appendix F, Sparse Crossing gives a model from which we decode a candidate Hamiltonian cycle by declaring edge E_k to belong to P whenever $E_k \leq P$. Figures 2h,i show in red the Hamiltonian cycle found for the difficult G7 graph from the FHCP challenge set [35] and for a random graph, respectively; additional examples are given in Appendix F.

Task axioms can also combine data and formal knowledge. For classification, we can use axioms to explain that numeric values are ordered. A natural encoding of numbers $t \in \{0, \ldots, L-1\}$ is an increasing chain $l(0) \leq l(1) \leq \cdots \leq l(L-1)$. We could then represent the number t in a term using the constant $l(t)$. However, this does not identify t uniquely: any larger value $t' > t$ is also compatible with $l(t)$, since $l(t) \leq l(t')$. A decreasing chain, $g(0) \geq g(1) \geq \cdots \geq g(L-1)$, has the dual ambiguity: $g(t)$ is also compatible with any smaller value $t' < t$. We therefore used both chains and encoded number t as the pair $l(t) \odot g(t)$. These two constants jointly pin down the number t by ruling out both higher and lower intensity values. We used this encoding for images to explain pixel intensity for each color channel, and for numerical variables in tabular data. Missing values in tabular data are simply missing in the term representing the table.

Adding this formal knowledge is necessary because otherwise AML would treat numerical values as unrelated symbolic constants. Information learned for one value would not automatically constrain nearby larger or smaller values. This is especially important when values appearing at test time are absent in the training data. See a controlled ablation study of formal knowledge in Appendix E.11.

We evaluated 24 datasets: 12 image datasets, including Fashion-MNIST [36], CIFAR-10 [31], and several medical datasets [32]; and 12 tabular datasets, including bank-marketing [37], Otto-product [38], Amazon employee access [39], Seismic-bumps [40], Gesture Phase Segmentation [33], First-order theorem proving [34], Philippine and Sylvine datasets [41], Phoneme [42], and Satellite [43]. Figures 2d,e extend the MNIST analysis in Figure 2a to CIFAR-10 and a skin lesion dataset from [32] as representative color-image datasets, while Figures 2f,g show two representative tabular dataset examples.

Test accuracy and macro-averaged F1 for the 24 classification datasets are reported in Table S10 for images and Table S11 for tabular data. Across the 24 datasets, AML without validation or cross-validation was statistically indistinguishable from tuned MLPs selected by validation or cross-validation, with two-sided paired Wilcoxon signed-rank tests giving $p = 0.4$ for both accuracy and macro-averaged F1. AML followed by logistic regression, also without validation or cross-validation, outperformed tuned MLPs, with $p = 0.01$ for accuracy and $p = 4.8 \cdot 10^{-4}$ for macro-averaged F1. The corresponding Hodges–Lehmann effect estimates [44] were $+1.9$ accuracy points, 95% CI $[0.4, 4.1]$, and $+0.04$ macro-averaged F1, 95% CI $[0.02, 0.06]$.

5 Discussion

Our results establish subdirect decomposition as a constructive principle for learning. In the instantiation studied here, tasks are specified by semilattice axioms, and learning consists of selecting subdirect components of the freest model. The theory explains why this selection can generalize: with sufficient data, the freest atomized model recovers the atoms of the hidden rule, while the finite-data analysis identifies which components are expected to have low test error. Sparse Crossing implements this selection principle as a practical algorithm.

Empirically, this mechanism supports practical generalization across 24 image and tabular datasets. The encodings are simple, no task-dependent hyperparameters are tuned, and no validation or cross-validation is used. Nevertheless, Sparse Crossing is competitive with tuned multilayer perceptrons selected by validation or cross-validation.

A distinctive feature of AML is that data and formal knowledge enter through the task axioms. This connects prediction tasks with purely formal problems, since the same machinery can learn from examples or build models satisfying formal specifications.

6 Limitations

The theoretical analysis is limited to atomized semilattices. This gives a constructive form of subdirect decomposition, but represents only one point in a broader algebraic design space. Other algebras may support richer encodings, smaller decompositions, or more efficient algorithms.

The method depends on the task encoding. The encodings used here demonstrate useful generalization from algebraic decomposition, but performance may depend on the encoding.

The probabilistic analysis allowed us to understand which atoms should be part of a generalizing model and to design Sparse Crossing and its subset-selection rule. However, this analysis does not characterize arbitrary subsets of atoms in the freest model where correlations between atom errors, label noise, or distribution shift may become relevant.

The finding of no overfitting and no need for validation datasets is empirical, and thus limited to our settings.

The implementation is not optimized for runtime efficiency and memory use (see present performance in Appendix E.5). Images larger than the ones used in the paper will require efficiency improvements through alternative encodings, algorithmic acceleration, parallelization, or dedicated hardware implementations.

The empirical comparison is limited to multilayer perceptrons as generic parametric learners (but see Tables S10 and S11 for a comparison against CNNs and XGBoost). This choice provides a baseline without task-specific inductive biases. A proper comparison with approaches introducing task-specific inductive biases would require to also incorporate domain structure into AML, for example through different task axioms or modified component-selection criteria.

Author Contributions

FMM and GP conceptualized the study, conducted the AML research, and wrote the manuscript. FMM and NA developed the open-source engine. DM performed all statistical analyses and conducted additional AML computations in response to reviewers' comments.

Acknowledgments

We thank Antonio Ricciardo, Emilio Suarez Canedo and Lars Wagoner for discussions. We are grateful for the support from Champalimaud Foundation (Lisbon, Portugal), from Portuguese national funding through FCT in the context of the project UIDB/04443/2025, and from the European Commission provided through projects H2020 ICT48 *Humane AI; Toward AI Systems That Augment and Empower Humans by Understanding Us, our Society and the World Around Us* (grant #820437) and the H2020 ICT48 project *ALMA: Human Centric Algebraic Machine Learning* (grant #952091).

References

[1] Ronald N. Bracewell. *The Fourier Transform and Its Applications*. McGraw-Hill, 3 edition, 2000.

[2] Elias M. Stein and Rami Shakarchi. *Fourier Analysis: An Introduction*. Princeton University Press, 2003.

[3] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge, 2 edition, 2013.

[4] Sheldon Axler. *Linear Algebra Done Right*. Springer, 3 edition, 2015.

[5] Garrett Birkhoff. Subdirect unions in universal algebra. *Bulletin of the American Mathematical Society*, 50(10):764–768, 1944.

[6] Stanley. Burris and H. P. Sankappanavar. *A course in universal algebra*. Springer-Verlag, 1981.

[7] Fernando Martin-Maroto and Gonzalo G. de Polavieja. Algebraic machine learning. *arXiv:1803.05252*, 2018.

[8] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[9] Fernando Martin-Maroto and Gonzalo G de Polavieja. Finite atomized semilattices. *arXiv:2102.08050*, 2021.

[10] Fernando Martin-Maroto and Gonzalo G de Polavieja. Semantic embeddings in semilattices. *arXiv:2205.12618*, 2022.

[11] Allen Newell and Herbert A. Simon. Computer science as empirical inquiry: Symbols and search. *Communications of the ACM*, 19(3):113–126, 1976.

[12] Edward A. Feigenbaum. The simulation of verbal learning behavior. In *Papers Presented at the May 9-11, 1961, Western Joint IRE-AIEE-ACM Computer Conference*, IRE-AIEE-ACM '61 (Western), page 121–132, New York, NY, USA, 1961. Association for Computing Machinery.

[13] Donald Michie. *Machine Learning, Neural and Statistical Classification*. Ellis Horwood, 1984.

[14] John F. Sowa. *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Brooks/Cole, 2000.

[15] Tim Rocktäschel and Sebastian Riedel. End-to-end differentiable proving. In *Advances in Neural Information Processing Systems*, 2017.

[16] Ryan Riegel, Fred Morstatter, Matthias Schaub, et al. Logical neural networks. *arXiv preprint arXiv:2006.13155*, 2020.

[17] Samy Badreddine, Artur d'Avila Garcez, Luciano Serafini, and Michael Spranger. Logic tensor networks. *Artificial Intelligence*, 303:103649, 2022.

[18] Bhabesh P. Bhuyan et al. Neuro-symbolic artificial intelligence: a survey. *Neural Computing and Applications*, 2024.

[19] Stephen Muggleton. Inductive logic programming. *New Generation Computing*, 8(4):295–318, 1991.

[20] Bernhard Ganter and Rudolf Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer, 1999.

[21] B. A. Davey and H. A. Priestley. *Introduction to Lattices and Order*. Cambridge University Press, Cambridge, 2 edition, 2002.

[22] Claudio Carpineto and Giovanni Romano. *Concept Data Analysis: Theory and Applications*. John Wiley & Sons, Hoboken, NJ, 2004.

[23] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International Conference on Machine Learning*, 2016.

[24] Irina Higgins, Mario Alberto Herrera, et al. Symmetry-based representations for artificial and biological general intelligence. *Frontiers in Computational Neuroscience*, 16:836498, 2022.

[25] Mathias Drton, Bernd Sturmfels, and Seth Sullivant. *Lectures on Algebraic Statistics*. Birkhäuser, 2009.

[26] Brendan Fong and David I. Spivak. *Seven Sketches in Compositionality: An Invitation to Applied Category Theory*. Cambridge University Press, 2018.

[27] Dan Shiebler. Category theory in machine learning. In *Applied Category Theory Conference*, 2021.

[28] Klaus Denecke and Shelly L Wismath. *Universal algebra and applications in theoretical computer science*. Chapman and Hall/CRC, 2018.

[29] Charles C Pinter. *A book of abstract algebra*. Courier Corporation, 2010.

[30] J. B. Nation. Notes on lattice theory. https://www.math.hawaii.edu/~jb/books.html.

[31] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[32] 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.

[33] Renata C. B. Madeo, Clodoaldo A. M. Lima, and Sarajane M. Peres. Gesture unit segmentation using support vector machines: segmenting gestures from rest positions. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, SAC '13, page 46–52, New York, NY, USA, 2013. Association for Computing Machinery.

[34] James Bridge, Sean Holden, and Lawrence Paulson. First-order theorem proving. UCI Machine Learning Repository, 2012. DOI: https://doi.org/10.24432/C5RC9X.

[35] M. Haythorpe. Fhcp challenge set: The first set of structurally difficult instances of the hamiltonian cycle problem. *Bulletin of the ICA, 83, 98-107*, 2018.

[36] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

[37] Sérgio Moro, P. Cortez, and Paulo Rita. A data-driven approach to predict the success of bank telemarketing. *Decis. Support Syst.*, 62:22–31, 2014.

[38] Benjamin Bossan, Josef Feigl, and Wendy Kan. Otto group product classification challenge. https://kaggle.com/competitions/otto-group-product-classification-challenge, 2015. Kaggle.

[39] Ben Hamner, kenmonta, and Will Cukierski. Amazon.com - employee access challenge. https://kaggle.com/competitions/amazon-employee-access-challenge, 2013. Kaggle.

[40] Marek Sikora and Lukasz Wrobel. seismic-bumps. UCI Machine Learning Repository, 2010. DOI: https://doi.org/10.24432/C5W902.

[41] Isabelle Guyon, Kristin Bennett, Gavin Cawley, Hugo Jair Escalante, Sergio Escalera, Tin Kam Ho, Núria Macià, Bisakha Ray, Mehreen Saeed, Alexander Statnikov, and Evelyne Viegas. Design of the 2015 chalearn automl challenge. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2015.

[42] Dominique Van Cappel. phoneme. OpenML Dataset Repository, 2015. OpenML ID: 1489.

[43] Markus Goldstein. Unsupervised Anomaly Detection Benchmark, 2015.

[44] Myles Hollander, Douglas A. Wolfe, and Eric Chicken. *Nonparametric Statistical Methods*. Wiley, July 2015.

[45] Imane M. Haidar, Layth Sliman, Issam W. Damaj, and Ali M. Haidar. Legacy versus algebraic machine learning: A comparative study. 2024.

[46] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*, 2015. arXiv:1412.6980.

[47] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[48] Richard Liaw, Eric Liang, Robert Nishihara, Philipp Moritz, Joseph E. Gonzalez, and Ion Stoica. Tune: A research platform for distributed model selection and training, 2018.

[49] Pouya Baniasadi, Vladimir Ejov, Jerzy A. Filar, Michael Haythorpe, and Serguei Rossomakhine. Deterministic "snakes and ladders" heuristic for the hamiltonian cycle problem. *Mathematical Programming Computation (2014) 6:55–75*, 2014.

Appendices

- **Appendix A**: **Atomized semilattices**
  Appendix A.2 adapts basic results from [9] to the machine-learning setting. Appendix A.3 provides new results that support the main text and the remaining appendices. Appendix A.1 collects relevant definitions for reference.
- **Appendix B**: **Discovery of rules in data**
  We give the theorems related to the discovery of underlying rules in data in AML.
- **Appendix C**: **Generalizing subsets of the freest atomized model**
  We give a probabilistic analysis of the expected false positive and negative ratios in the freest atomized model.
- **Appendix D**: **Sparse crossing**
  We give an in-depth analysis of the Sparse Crossing algorithm, including pseudocode.
- **Appendix E**: **Methods**
  We give details of the methods used in the empirical tests.
- **Appendix F**: **Formal problems**
  We show how to semantically embed a Hamiltonian cycle problem and show the solutions found by Full Crossing in a simple case and by Sparse Crossing in complex cases.
- **Appendix G**: **Comparison of accuracies**
  We give the accuracies and F1 scores of AML and multilayer perceptrons for 12 image datasets and 12 tabular datasets

A Atomized Semilattices

We provide in Appendix A.2 an adaptation some basic results from [9] relevant to the context of machine learning, and we give new results in atomized semilattices in Appendix A.3 to support the main text and other Appendices. Appendix A.1 gives relevant definitions for reference.

A.1 Definitions

Definition 1. A **semilattice** is an algebra with a single binary function $\odot$, that satisfies the commutative, associative and idempotent properties: $\forall x \forall y \ [(x \odot y) = (y \odot x)]$, $\forall x \forall y \forall z [x \odot (y \odot z) = (x \odot y) \odot z]$ and $\forall x \ [(x \odot x) = x]$

Definition 2. The **component constants** of a single constant c is defined as the constant itself, $\mathbf{C}(c) = \{c\}$, and the component constants of a term $t = c_1 \odot c_2 \odot ... \odot c_n$ as the set $\mathbf{C}(t) = \{c_1, c_2, ..., c_n\}$.

Definition 3. Positive and negative duples in a model. We use the word *duple* to refer to an ordered pair of terms, $r \equiv (a, b)$. This notation is silent about whether it is valid or not in a model. We say that a duple is positive in a model M, and we denote it by r^+, if the duple is valid in the model, $M \models (a \leq b)$. Similarly, we say a duple is negative, and write r^-, if $M \not\models (a \leq b)$ or, equivalently, $M \models (a \not\leq b)$. For a set of positive duples we use the notation R^+, with $R^+ = \{r_1^+, r_2^+, ..., r_N^+\}$ and for a set of negative duples of positive duples we use the notation R^-, with $R^- = \{r_1^-, r_2^-, ..., r_N^-\}$.

Definition 4. Theory of a model. For any model M, its theory, denoted $Th(M)$, is the set of sentences (duples) satisfied by M. The subscript "0" in $Th_0(M)$ specializes sentences without quantifiers, i.e. atomic and negative atomic sentences. We use $Th_0(M^+)$ and $Th_0(M^-)$ when we want to refer to all the positive or negative duples satisfied by M, respectively.

Definition 5. A semilattice M_1 is **freer than or as free** as the semilattice M_2 if for every duple r for which $M_2 \models r^-$ we also have $M_1 \models r^-$. Equivalently, $Th_0^-(M_2) \subseteq Th_0^-(M_1)$.

Definition 6. The **freest model** over the constants C of the set of positive duples R^+, $F_C(R^+)$, is the model such that if $F_C(R^+) \models r^+$ for any duple r then all other models of R^+ also satisfy r^+.

Definition 7. An **atomized semilattice** over a set of constants C is a structure M with elements of two sorts, the regular elements in Latin letters $\{a, b, c, ...\}$ and the atoms in Greek letters $\{\phi, \psi, ...\}$, with an idempotent, commutative and associative binary operator $\odot$ defined for regular elements and a partial order relation $<$ (i.e. a binary, reflexive, antisymmetric and transitive relation) that is defined for both regular elements and atoms, such that the regular elements are either constants or idempotent summations of constants, and M satisfies the axioms of the operations and the additional:

$$\forall \phi \exists c : (c \in C) \wedge (\phi < c), \tag{AS1}$$

$$\forall \phi \forall a \, (a \not\leq \phi), \tag{AS2}$$

$$\forall a \forall b \, (a \leq b \Leftrightarrow \neg \exists \phi : ((\phi < a) \wedge (\phi \not< b))), \tag{AS3}$$

$$\forall \phi \forall a \forall b \, (\phi < a \odot b \Leftrightarrow (\phi < a) \vee (\phi < b)), \tag{AS4}$$

$$\forall c \in C \ ((\phi < c) \Leftrightarrow (\psi < c)) \Rightarrow (\phi = \psi), \tag{AS5}$$

$$\forall a \exists \phi : (\phi < a). \tag{AS6}$$

Definition 8. An **atom** ϕ of an atomized semilattice M over C is determined by its upper constant segment, the set $U^c(\phi) \subseteq C$, as follows: $\phi <_M c$ if and only if $(\phi \in M) \wedge (c \in U^c(\phi))$.

Definition 9. The **lower atomic segment** of a regular element x of a model M atomized by a set of atoms A is $L_M^a(x) = \{\phi : (\phi < x) \wedge (\phi \in A)\}$.

Definition 10. The **upper constant segment** of an atom ϕ, in any model, is the constants in which the atom ϕ is in, $\mathbf{U}^c(\phi) = \{c : (\phi < c) \wedge (c \in C)\}$. Atoms are "universally" defined independently of a particular model as a consequence of Theorem 1 (iv) and Theorem 1 (iv).

Definition 11. We say that an atom ϕ is **wider than** an atom η if $U^c(\eta) \subsetneq U^c(\phi)$, with the upper constant segment U^c as in Definition 10.

Definition 12. Redundant atom. An atom ϕ is *redundant* in model M if and only if for each constant c such that $\phi < c$ there is at least one atom $\eta < c$ in M with ϕ larger than η, with the notion of larger as in Definition 11.

Definition 13. The **discriminant** of terms a and b in the atomized model M, written as $dis_M(a,b)$, is the set of atoms in a that are not in b. Using the definition of lower atomic segment in Definition 9, we can write it as $dis_M(a,b) = L^c_M(a) \setminus L^c_M(b)$.

Definition 14. The **union of atoms** ϕ and ψ is an atom represented as $\phi \bigtriangledown \psi$ with upper constant segment is $U^c(\phi \bigtriangledown \psi) = U^c(\phi) \cup U^c(\psi)$, with U^c the upper constant segment in Definition 10.

Definition 15. Full-crossing. Let M be our model and let $r \equiv (r_L, r_R)$ be a duple that is not valid in the model, $M \not\models r^+ \equiv (r_L < r_R)$. Let H be the discriminant of the terms r_L and r_R, $H = dis_M(r_L, r_R)$, with the discriminant in Definition 13. Let B be the set of atoms of M that are in r_R, $B = L^a_M(r_R)$ with L^a_M the lower atomic segment in Definition 9. The *full-crossing* of duple r into model M gives a new model $\square_r M$ (read as "the full-crossing of r into M") atomized by:

$$\square_r M = (M \setminus H) \cup (H \bigtriangledown B),$$

where $\square_r$ is the Full Crossing operator and $H \bigtriangledown B$ the set of atoms resulting from all pairwise unions of each atom of H and each atom of B,

$$H \bigtriangledown B \equiv \{\lambda \bigtriangledown \rho : (\lambda \in H) \wedge (\rho \in B)\},$$

with the atom union of atoms, $\lambda \bigtriangledown \rho$, in Definition 14.

Definition 16. Causal set. Let P and Q be two sets, each of positive and negative duples. P is a causal set of Q when every duple of Q is a logical consequence of the duples in P.

Definition 17. The **pinning term** of atom ϕ in a semilattice over the constants C is the term with component constants $C - U^c(\phi)$.

Definition 18. Let the null atom, $\ominus_c$, be the atom with every constant of C in its upper constant segment, i.e. $U^c(\ominus_C) = C$.

A.2 Basic results

Theorem 1. *Let $t, s \in F_C(\emptyset)$ be two terms that represent two regular elements $\nu_M(t)$ and $\nu_M(s)$ of an atomized model M over a finite set of constants C. Let ϕ be an atom, c a constant in C and let a be a regular element of M:*

i) $\forall t \forall c (c \in \mathbf{C}(t) \Rightarrow \nu_M(c) \leq \nu_M(t))$,

ii) $\phi < \nu_M(t) \Leftrightarrow \exists c : ((c \in \mathbf{C}(t)) \wedge (\phi < \nu_M(c)))$,

iii) $(\phi < a) \Leftrightarrow \exists c : ((c \in C) \wedge (\phi < \nu_M(c) \leq a))$,

iv) $L^a_M(\nu_M(t)) = \{\phi \in M : \mathbf{C}(t) \cap U^c(\phi) \neq \emptyset\}$,

v) $L^a_M(\nu_M(s) \odot \nu_M(t)) = L^a_M(\nu_M(t)) \cup L^a_M(\nu_M(s))$,

vi) $\nu_M(t) \leq \nu_M(s) \Leftrightarrow L^a_M(\nu_M(t)) \subseteq L^a_M(\nu_M(s))$.

Proof. (i) From $t = t \odot c$ and the natural homomorphism $\nu_M(t) = \nu_M(t \odot c) = \nu_M(t) \odot \nu_M(c)$ we get $\nu_M(c) \leq \nu_M(t)$.
(ii) Right to left, $\phi < \nu_M(c) \leq \nu_M(t)$ follows from (i) and, from and the transitivity of the order relation, $\phi < \nu_M(t)$. Left to right can be proven from the fourth axiom of atomized models $\phi < a \odot b \Rightarrow (\phi < a) \vee (\phi < b)$ applied to the component constants $\mathbf{C}(t)$ of t. The number of component constants of t is at least 1 and at most $|C|$ so it is a finite number and we need to apply this axiom a finite number of times to get $\phi < \nu_M(c)$ for some component constant of t. This proves (ii), and (iii) is a consequence of (ii) that follows with just choosing any term t of $F_C(\emptyset)$ to represent element $a = \nu_M(t)$.
(iv) Consider an atom $\phi \in L^a_M(\nu_M(t))$ then $\phi < \nu_M(t)$ and from proposition (ii) there is a component constant $c \in \mathbf{C}(t)$ such that $\phi < \nu_M(c)$ which means that $c \in \mathbf{C}(t) \cap U^c(\phi)$. Conversely, if $c \in \mathbf{C}(t) \cap U^c(\phi)$ then $\phi < \nu_M(c) \leq \nu_M(t)$ and $\phi \in L^a_M(\nu_M(t))$.
(v) Since ν_M is a homomorphism $\nu_M(s) \odot \nu_M(t) = \nu_M(s \odot t)$ and, using proposition (iv) $L^a_M(\nu_M(s \odot t)) = \{\phi \in M : \mathbf{C}(s \odot t) \cap U^c(\phi) \neq \emptyset\} = \{\phi \in M : (\mathbf{C}(s) \cup \mathbf{C}(t)) \cap U^c(\phi) \neq \emptyset\} = L^a_M(\nu_M(t)) \cup L^a_M(\nu_M(s))$. Note that this proposition is an alternative way to write axiom AS4.
(vi) It is straightforward from proposition (v) and AS3. □

Theorem 2. *Assume the axiom AS4 of the atomized semilattice and the antisymmetry of the order relation.*
i) AS3 implies AS3b, with AS3 an axiom of the atomized semilattice and AS3b the partial order of the semilattice:

$$\forall a \forall b \, (a \leq b \;\Leftrightarrow\; a \odot b = b). \tag{AS3b}$$

ii) Assume $\forall a \forall b (\forall \phi ((\phi < a) \Leftrightarrow (\phi < b)) \Rightarrow (a = b))$. *Then* $AS3b \Rightarrow AS3$.
iii) AS3 implies $\forall a \forall b (\forall \phi ((\phi < a) \Leftrightarrow (\phi < b)) \Rightarrow (a = b))$.

Proof. (i) $AS3 \Rightarrow AS3b$: Assume $a \leq b$. AS3 implies $\neg \exists \phi : ((\phi < a) \wedge (\phi \not< b))$ and then $(\phi < a) \vee (\phi < b) \Leftrightarrow (\phi < b)$. From here, using AS4, follows $(\phi < a \odot b) \Leftrightarrow (\phi < b)$, and we can use AS3 (right to left this time) to get $(b \leq a \odot b) \wedge (a \odot b \leq b)$, and from the antisymmetry of the order relation, i.e. $(x = y) \Leftrightarrow (x \leq y) \wedge (y \leq x)$, we obtain $a = a \odot b$.
Assume $a \odot b = b$. Then $(\phi < a \odot b) \Leftrightarrow (\phi < b)$, and using AS4: $(\phi < a) \vee (\phi < b) \Leftrightarrow (\phi < b)$ which implies $\neg \exists \phi : ((\phi < a) \wedge (\phi \not< b))$ from which, using AS3, we get $a \leq b$.
(ii) $AS3b \Rightarrow AS3$: Assume $a \leq b$. AS3b implies $a \odot b = b$. From AS4 we get that $(\phi < a) \vee (\phi < b) \Leftrightarrow (\phi < a \odot b = b)$, so $\phi < a$ implies $\phi < b$ and then $\neg \exists \phi : ((\phi < a) \wedge (\phi \not< b))$.
Assume now that $\neg \exists \phi : ((\phi < a) \wedge (\phi \not< b))$. Then $(\phi < a) \vee (\phi < b) \Leftrightarrow (\phi < b)$ and AS4 leads to $(\phi < a \odot b) \Leftrightarrow (\phi < b)$. However, we cannot go from here to $a \leq b$ unless we add an axiom such as: $\forall \phi ((\phi < a) \Leftrightarrow (\phi < b)) \Rightarrow (a = b)$.
(iii) From $\forall \phi ((\phi < a) \Leftrightarrow (\phi < b))$ follows, using AS3, that $(a \leq b) \wedge (b \leq a)$, and by the antisymmetry of the order relation we get $a = b$. □

Theorem 3. *Let* $t, s \in F_C(\emptyset)$ *be two terms. If an atom* ϕ *of a model* M *discriminates a duple* (t, s) *in* M *then it discriminates* (t, s) *in any model that contains* ϕ.

Proof. Consider two atomized models M and N that contain ϕ. If ϕ discriminates a duple (t, s) in M from part (ii) of Theorem 4 follows that ϕ discriminates (t, s) in $F_C(\emptyset)$ and then using part (ii) again ϕ also discriminates (t, s) in N. □

Theorem 4. *Let* $t, s \in F_C(\emptyset)$ *be two terms and* M *an atomized model. Let* ϕ *be an atom and* $\nu_M : F_C(\emptyset) \to M$ *the natural homomorphism.*

$$i) \;\; (F_C(\emptyset) \models (t \leq s)) \;\Rightarrow\; M \models (\nu_M(t) \leq \nu_M(s))$$

$$ii) \;\; (\phi \in M) \wedge (F_C(\emptyset) \models (\phi < s)) \;\Leftrightarrow\; M \models (\phi < \nu_M(s))$$

Proof. i) This proposition is a well-known fact that follows from the fact that ν_M is a homomorphism. Proposition (i) is provided here for comparison with proposition (ii).
ii) Note that we use here the same atom ϕ in the contexts of two atomized models, $F_C(\emptyset)$ and M. Left to right: using Theorem 1 (iii), $F_C(\emptyset) \models (\phi < s)$ implies that there is some constant c such that $F_C(\emptyset) \models (\phi < c \leq s)$ and then, from the natural homomorphism, $M \models (\nu_M(c) \leq \nu_M(s))$. From Definition 8, $F_C(\emptyset) \models (\phi < c)$ requires $c \in U^c(\phi)$ and then, because we assume $\phi \in M$, the same definition allows us to write $M \models (\phi < \nu_M(c))$. Using the transitive property of the order relation $M \models (\phi < \nu_M(s))$.
Right to left is essentially the same proof as left to right except for the fact that we do not need to require $\phi \in F_C(\emptyset)$ as this is always true for any atom. □

Theorem 5. *An atom* ϕ *of an atomized model* M *can be eliminated without altering the model if and only if* ϕ *is redundant.*

Proof. Let $Th^+(M)$ be the set of all positive duples satisfied by the regular elements of M (all elements are regular except the atoms). Since positive duples do not become negative when atoms are eliminated, taking out an atom ϕ from a model M produces a model N of $Th^+(M)$. Therefore, when removing an atom we only need to worry about negative duples that may become positive. To prove that a redundant atom can be eliminated let a and b be a pair of regular elements and $(a \not\leq b)$ a negative duple satisfied by M and discriminated by an atom $\phi < c \leq a$ where c is some constant.

If ϕ is redundant there is an atom $\eta < c$ in M such that ϕ is larger than η. Suppose $\eta < b$. There is a constant e such that $\eta < e \leq b$. Because ϕ is larger than η then $\phi < e \leq b$ should also hold contradicting the assumption that $\phi \in dis_M(a, b)$. We have proved that $N \models (\eta \not< b)$. Therefore, any negative duple of M is also satisfied by N. If N models the same positive and negative duples than M then the subalgebras of M and N spawned by the regular elements are isomorphic.
Conversely, assume atom ϕ can be eliminated without altering M. The pinning term T_ϕ is the idempotent summation of all the constants in the set $C \setminus U^c(\phi)$. For each constant c such that $\phi < c$ we have $\phi \in dis_M(c, T_\phi)$. That ϕ discriminates duple (c, T_ϕ) follows from Theorem 3, the fact that c and T_ϕ are terms and $F_C(\emptyset) \models (\phi \not< T_\phi)$. If ϕ can be eliminated without altering M, for each constant c in $U^c(\phi)$ there should be some atom $\eta_c < c$ with $\eta_c \neq \phi$ in M discriminating the duple (c, T_ϕ) which implies $T_\phi < T_{\eta_c}$ or, equivalently, that ϕ is larger than η_c. Hence, for each constant such $\phi < c$ there is an $\eta_c \in M$ such that ϕ is larger than η_c which proves ϕ is redundant. □

Theorem 6. *Let M be a model and ϕ a non-redundant atom of M such that there is at least one constant in C that is not in the upper constant segment of ϕ. There is at least one pinning duple that is discriminated by ϕ and only by ϕ.*

Proof. As in the proof of Theorem 5, for each constant $c \in U^c(\phi)$ we have $\phi \in dis_M(c, T_\phi)$ where T_ϕ is the pinning term of ϕ. This is a consequence of Theorem 3 and $F_C(\emptyset) \models (\phi \not< T_\phi)$. If, for a pinning duple (c, T_ϕ) there is another atom φ of M that discriminates (c, T_ϕ) then ϕ is larger than φ and, if the same is true for every pinning duple, then ϕ is redundant with M, which is against our assumptions. Therefore, there should be at least one constant c such that (c, T_ϕ) is discriminated only by ϕ. □

Theorem 7. *Let x be a regular element of a model M and let ϕ, ψ and η be atoms of M. The union of atoms has the properties:*

i) $\phi \bigtriangledown \phi = \phi$,

ii) $\bigtriangledown$ *is commutative and associative,*

iii) $\phi < x \Rightarrow (\phi \bigtriangledown \psi < x)$,

iv) $(\phi \bigtriangledown \psi < t) \Leftrightarrow (\phi < x) \vee (\psi < x)$,

v) $(\phi \bigtriangledown \psi < x) \wedge (\phi \not< x) \Rightarrow (\psi < x)$.

vi) ϕ *is larger or equal to* η *if and only if* $\phi = \phi \bigtriangledown \eta$.

Proof. An atom is determined by the constants in its upper segment, therefore atom $\phi \bigtriangledown \psi$ is fully defined by $U^c(\phi \bigtriangledown \psi) = U^c(\phi) \cup U^c(\psi)$ and then (i) and (ii) follow from the idempotence, commutativity and associativity of the union of sets. Choose a term $t \in F_C(\emptyset)$ such $x = \nu(t)$ with ν the natural homomorphism of $F_C(\emptyset)$ onto M. From Theorem 1 we know

$$L^a_M(\nu(t)) = \{\phi \in M : \mathbf{C}(t) \cap U^c(\phi) \neq \emptyset\},$$

which shows how to calculate the lower atomic segment of an element represented with any term by using the component constants of the term. We can use this duple to prove the other statements. $\phi < x$ implies that exists $c \in \mathbf{C}(t)$ such that $x = \nu(t)$ and $\phi < c$, hence, $c \in U^c(\phi \bigtriangledown \psi)$ so $\phi \bigtriangledown \psi < c \leq x$ and (iii) follows. (iv) right to left says that there is a constant $c \in \mathbf{C}(t)$ and a term t such that $x = \nu(t)$ and $(\phi < c) \vee (\psi < c)$ which implies $\phi \bigtriangledown \psi < c \leq x$. To prove (iv) left to right, write the right side as $\exists t \exists c(x = \nu(t) \wedge c \in \mathbf{C}(t))$ such that $\phi \bigtriangledown \psi < c$, which implies $c \in U^c(\phi) \cup U^c(\psi)$ and then $(\phi < c) \vee (\psi < c)$ that, together with $c \leq x$, yields $(\phi < x) \vee (\psi < x)$. (v) can be proved in the same way then the others and is left to the reader. (vi) ϕ is larger than, or equal to, atom η if and only if for every constant c, $(\eta < c) \Rightarrow (\phi < c)$ or, in other words, $U^c(\eta) \subseteq U^c(\phi)$. It follows that $U^c(\phi \bigtriangledown \eta) = U^c(\phi) \cup U^c(\eta) = U^c(\phi)$. Hence, ϕ is larger than or equal to atom η if and only if $\phi = \phi \bigtriangledown \eta$. □

Theorem 8. *Let M be an atomized model over a finite set C of constants. Let ϕ be an atom that may or may not be in the atomization of M.*
i) ϕ is redundant with M if and only if it is a union $\phi = \bigtriangledown_i \eta_i$ of atoms of M such that $\forall i(\phi \neq \eta_i)$.
ii) ϕ is redundant with M if and only if it is a union of two or more non-redundant atoms of M.

Proof. If ϕ is redundant, for each constant such that $\phi < c$ there is an atom η_i of M such that ϕ is larger than η_i and $\eta_i < c$. Theorem 7 assures us that if ϕ is larger than η_i then $\phi = \phi \bigtriangledown \eta_i$ and, since for each constant $c \in U^c(\phi)$ there is some η_i such that $\eta_i < c$ then $U^c(\phi) = \cup_i U^c(\eta_i)$. It follows $\phi = \bigtriangledown_i \eta_i$. Conversely if ϕ is a union of atoms $\phi = \bigtriangledown_i \eta_i$ then for each constant such $\phi < c$ there is some atom η_i that contains c in its upper constant segment with ϕ larger than η_i, hence, ϕ is redundant. This proves (i).
Since C is finite then $|U^c(\eta_i)| < |U^c(\phi)|$. If any of the atoms η_i is redundant with M it can be further expressed as unions of atoms of M with ever smaller upper constant segments until reaching non-redundant atoms of M. Because $\bigtriangledown$ is associative there is at least one decomposition of ϕ as a union of non-redundant atoms of M. □

Theorem 9. *i) Two atomizations of the same model have the same non-redundant atoms.*
ii) Any model has a unique atomization without redundant atoms.

Proof. Let A and B be two atomizations of a model M without redundant atoms. Choose an atom ϕ of B and consider the model $A + \{\phi\}$ spawned by ϕ and the atoms of A. It is clear that $A + \{\phi\}$ spawns the same model as A otherwise there is a positive duple of A discriminated only by ϕ and, hence, negative in B contradicting that A and B are atomizations of the same model. From Theorem 5 either ϕ is an atom of A or ϕ is redundant with A. Assume ϕ is redundant with A. There is a set E_ϕ of atoms of A such that ϕ is a union of the atoms in E_ϕ (see Theorem 8). Choose an atom η in E_ϕ and consider the model $B + \{\eta\}$. The same reasoning applies so we should get that either η is in B or is redundant with atoms of B. We can substitute η in E_ϕ with the atoms that make η redundant in B to form a set E'_ϕ. In this way we can replace every atom of E_ϕ with atoms of B such that ϕ is a union of the atoms in E'_ϕ which implies that ϕ is redundant in B, against our assumptions. Therefore, ϕ cannot be redundant with A and then ϕ should be an atom of A, which proves proposition (i) and also proposition (ii) because A and B should be identical. □

Theorem 10. *Let A and B be two atomized models, with A freer or as free as B.*
i) The model $A + B$ spawned by the atoms of A and the atoms of B is the same as the model spawned by A alone.
ii) The atoms of B are in A or are redundant with the atoms of A.
iii) A is freer than B if and only if the atoms of B are atoms of A or unions of atoms of A.

Proof. (i) Since A is freer or as free as B all the negative duples of B are also negative in A. This means that a duple discriminated by an atom of B is also discriminated by some atom of A. In addition, each positive duple of A is also a positive duple of B and also a positive duple of $A + B$. A duple of $A + B$ is positive if and only if it is positive in A, and is negative if and only if it is negative in A. Therefore, the models $A + B$ and A are equal. Part (ii) follows directly from Theorem 9 and the fact that $A + B$ spawns the same model as A.
(iii) Assume A is freer than B. Consider the model $A + B$ spawned by the atoms of A and the atoms of B. Proposition (i) tells us that if A is freer than B the model $A + B$ is equal to A. Theorem 5 assures us that each atom ϕ of B is an atom of A or is redundant with the atoms of A. Either ϕ is an atom of A or for each constant c such that $\phi < c$ there is at least one atom $\eta < c$ in A such that ϕ is larger than η, i.e. the set $U^c(\phi)$ contains the set $U^c(\eta)$. In fact, as there is an η for each constant $c \in U^c(\phi)$, if the set $\{\eta_i : i = 1, ..., n\}$ makes ϕ redundant in N, then $U^c(\phi) = \cup_i U^c(\eta_i)$ and then ϕ is the union $\phi = \bigtriangledown_i \eta_i$.
Assume now that the atoms of B are atoms of A or unions of atoms of A. All the atoms of B are redundant with A and then, Theorem 5 assures that any duple r discriminated by an atom of B is discriminated by at least one atom of A so, $B \models r^-$ implies $A \models r^-$ and A is freer or equal to B. □

Theorem 11. *Let M be an atomized model with or without redundant atoms and r a duple so $M \models r^-$. The full-crossing of r in M is the freest model $F_C(Th^+(M) \cup r^+)$.*

Proof. Let $H \subseteq M$ be the discriminant of $r = (r_L, r_R)$, i.e. the set of atoms φ such that $\varphi < r_L \wedge \varphi \not< r_R$. Let $B \subseteq M$ be the set of atoms of r_R, i.e. the atoms φ such that $\varphi < r_R$. The full-crossing of (r_L, r_R) in model M is the model $K = (M \setminus H) \cup (H \bigtriangledown B)$ where $H \bigtriangledown B \equiv \{\lambda \bigtriangledown \rho : (\lambda \in H) \wedge (\rho \in B)\}$ is the set of all pairwise unions of an atom of H and an atom of B.

Using Theorem 7 (iii) it follows that the atoms $\lambda \bigtriangledown \rho \in H \bigtriangledown B \subseteq K$ introduced by the full-crossing operation satisfy $\lambda \bigtriangledown \rho < r_R$ because $\rho < r_R$. Since the atoms in the discriminant $\lambda \in H = dis_M(r)$ are no longer present in K and the atoms introduced by the full-crossing are all in the lower segment of r_R then $K \models r^+$.

It is immediate from the definition of the order relationship, $<$, in atomized models (the axiom $\forall a \forall b\, (a \leq b \Leftrightarrow \neg\exists\phi((\phi < a) \wedge (\phi \not< b)))$ that the elimination of atoms from a model cannot cause any positive duple to become negative. Hence, the atoms eliminated by the full-crossing operation cannot switch positive duples into negative. We have to prove that the atoms introduced by the full-crossing do not switch positive duples into negative duples either. Assume $M \models s^+$ for some duple $s = (s_L, s_R)$. Suppose that s_L acquires one of the new atoms $\lambda \bigtriangledown \rho$ in its lower segment. From Theorem 7 (iv) follows that either $\lambda < s_L$ or $\rho < s_L$ holds in M. Because $s_L \leq s_R$ in M then $M \models (\lambda < s_R) \vee (\rho < s_R)$. By Theorem 7 (iii) we get $K \models (\lambda \bigtriangledown \rho < s_R)$. Therefore, the atoms of the form $\lambda \bigtriangledown \rho$ cannot switch a positive duple s^+ into s^-.

So far we know that $K \models r^+$ and that $(M \models s^+) \Rightarrow (K \models s^+)$ so we can write: $K \models Th^+(M) \cup r^+$. To prove that K is the freest model of $Th^+(M) \cup r^+$ we have to show that the full-crossing does not switch negative duples into positive either unless they are logical consequences of $Th^+(M) \cup r^+$.

Consider a duple $s = (s_L, s_R)$ such that $K \models s^+$ and $M \models s^-$. The crossing of r has switched s from negative to positive. For this to occur a necessary condition is that the discriminant of s should disappear from K as a result of the crossing, in other words, $dis_M(s) \subseteq H = dis_M(r)$, so the atoms of $dis_M(s)$ should all be atoms of r_L. This implies $s_L \leq s_R \odot r_L$ holds in M.

Since $M \models s^-$ there is at least one $\lambda^* \in dis_M(s)$. From Theorem 7 (iii), all the atoms of the form $\{\lambda^* \bigtriangledown \rho : \rho \in B\}$ are in s_L. If $K \models s^+$ then all the atoms $\lambda^* \bigtriangledown \rho$ should also be atoms of s_R in model K. Using Theorem 7 (v), $\lambda^* \not< s_R$ and $\lambda^* \bigtriangledown \rho < s_R$ implies $\rho < s_R$ for each $\rho \in B$. Since B is the entire lower atomic segment of r_R in M it immediately follows that $M \models (r_R \leq s_R)$. Putting both conditions together:

$$M \models ((s_L \leq s_R \odot r_L) \wedge (r_R \leq s_R)).$$

Since

$$(s_L \leq s_R \odot r_L) \wedge (r_R \leq s_R) \wedge (r_L \leq r_R) \Rightarrow s_L \leq s_R$$

s^+ is a logical consequence of the theory $Th^+(M) \cup r^+$ so any model of $Th^+(M) \cup r^+$ must satisfy s^+. We have proved that any duple s^+ satisfied by K is satisfied by any model of $Th^+(M) \cup r^+$ and, because $K \models Th^+(M) \cup r^+$ then $K = F_C(Th^+(M) \cup r^+)$. □

Theorem 12. *The full-crossing operation is commutative up to redundant atoms, i.e. the resulting model of full-crossing each duple in a set of duples R is independent of the order chosen.*

Proof. Theorem 11 shows that the result of full-crossing the duples in a set R in a model M is the freest model $F_C(Th^+(M) \cup R^+)$, so the result is the same independently of the order of crossing. Because the result is the same model, from Theorem 5, the resulting atomizations for different orders of crossing can only differ in redundant atoms. □

Theorem 13. *The freest model $F_C(\emptyset)$ over a set C of constants has $|C|$ non-redundant atoms, each with a single constant in its upper segment.*

Proof. It is a well-known result of Universal Algebra that the freest model is the term algebra, i.e. the algebra spawned by the terms modulus the rules of the algebra, in this case the commutative, associative and idempotent laws. For terms s and t, the term algebra satisfies that $F_C(\emptyset) \models s \leq t$ if and only if the component constants satisfy $\mathbf{C}(s) \subseteq \mathbf{C}(t)$. Let M be the atomized model obtained by $|C|$ different atoms, each with one constant in its upper segment. From the axiom of atomized models $\phi < a \odot b \Leftrightarrow (\phi < a) \vee (\phi < b)$ applied to the component constants of s and t follows that for M (and for any atomized model), $\mathbf{C}(s) \subseteq \mathbf{C}(t)$ implies $s \leq t$. Conversely, assume $M \models s \leq t$. Each atom ϕ in the lower atomic segment of s should be in the lower segment of some component constant $c \in \mathbf{C}(s)$ (c.f. Theorem 1 (ii)) and, since the atoms of M have only one constant in their upper segments then $\phi < t$ can occur only if c is also a component constant of t. Since each constant of C has its own atom, every component constant of s should be a component constant of t otherwise there is an atom that discriminates (s, t) against our assumption. This proves that M models the exact same duples as $F_C(\emptyset)$ and both models are the same. Since each atom ϕ of M has an upper segment $U^c(\phi)$ with a single constant ϕ is non-redundant. We have finished the proof.
We can easily show that M is freer than any atomized model. Let N be any atomized model and let r be any duple such that $N \models r^-$. Each atom η of N is the union of the atoms of M corresponding to each constant in the set $U^c(\eta)$ and this proves that η is redundant with M. Since all the atoms of N are redundant with M, using Theorem 10 (iii) we conclude that M is freer or as free as N and, therefore, $M \models r^-$ and M is the freest model. From Theorem 5 any atomization of $F_C(\emptyset)$ contains the $|C|$ atoms of M and only can differ from M in a set of redundant atoms. □

Theorem 14. *Any model M with a finite set C of constants can be atomized.*

Proof. Full-crossing provides a simple constructive proof for semilattices. Since the theory $Th^+(M)$ of any model M over a finite set of constants is a finite a set, M can be atomized by starting with the set of $|C|$ atoms that provides an atomization for the freest model $F_C(\emptyset)$ (see Theorem 13), each atom contained in a single constant, and then by performing a finite sequence of full-crossing operations for each duple in $Th^+(M)$. As a result we obtain an atomization of model M which follows from Theorem 11 and $M = F_C(Th^+(M))$. □

Theorem 15. *Let M be an atomized model over a set C of constants.*
i) If M satisfies a negative duple r^- then there is a pinning duple $p = (c, T_\phi)$ for some constant $c \in U^c(\phi)$ and the pinning term T_ϕ such that $p^- \Rightarrow r^-$.
ii) Let $PR(M) \subset Th_0^-(M)$ be the set of pinning duples of M. $PR(M)$ implies $Th_0^-(M)$.
iii) Let α and β be two different atoms of a model M then $M \models (T_\alpha \neq T_\beta)$.

Proof. Let $r = (r_L, r_R)$. If $M \models r^-$ there is an atom ϕ of M that discriminates r^-. It follows that there is a constant c such that $\phi < c$ and a sentence $\Delta = (c \leq r_L) \wedge (r_R \leq T_\phi)$ that is true not only in M, but also in the freest model $F_C(\emptyset)$ and in every model with constants in C. Suppose a model N satisfies r^+; then, $N \models \Delta \wedge r^+$ and it follows that $c \leq r_L \leq r_R \leq T_\phi$ which implies p^+. In other words $p^- \Rightarrow r^-$. This proves proposition (i) from which (ii) follows directly.
(iii) Assume $M \models (T_\alpha = T_\beta)$. Since $\alpha \not< T_\alpha$ then we also must have $\alpha \not< T_\beta$, which implies $U^c(\alpha) \subseteq U^c(\beta)$. Using the symmetrical, $\beta \not< T_\beta$ we get $U^c(\beta) \subseteq U^c(\alpha)$. Hence, α and β are equal against our hypothesis. □

Theorem 16. *Let Γ be a set of first order sentences, R a set of positive and negative duples so $\Gamma \Rightarrow R$, and M a model of Γ. Any atom of M is an atom of $F(R^+)$ or is redundant with atoms of $F(R^+)$.*

Proof. Since Γ implies R any model M of Γ also satisfies R^+. Therefore, $F(R^+)$ is freer than M or equal to M. From Theorem 10, each atom of M is an atom of $F(R^+)$ or is redundant with atoms of $F(R^+)$. □

Theorem 17. *Let $R^+ \subset Th_0^+(M)$ be a subset of the positive theory of some atomized model M and let $F(R^+)$ be the freest model of R^+. The atoms of M are atoms of $F(R^+)$ or unions of atoms of $F(R^+)$.*

Proof. Follows from Theorem 10 (iii) and the fact that $F(R^+)$ is freer that M. □

Theorem 18. *Let M be an atomized semilattice with atomization A. For each atom $\phi \in A$ with $\phi \neq \ominus_c$ (see Definition 18), define*

$$h_\phi : M \longrightarrow \mathbf{2}, \qquad h_\phi(t) = \begin{cases} 1 & \text{if } \phi < t, \\ 0 & \text{if } \phi \not< t. \end{cases}$$

The map

$$\Phi : M \longrightarrow \prod_{\phi \in A,\, \phi \neq \ominus_c} \mathbf{2}, \qquad t \longmapsto \big(h_\phi(t)\big)_\phi$$

is a subdirect embedding of M into a power of the two-element semilattice.

Proof. Let $\mathbf{2} = \{0, 1\}$ with $0 < 1$ and idempotent sum $a \odot b = \max(a, b)$ (the unique non-trivial two-element semilattice). A subdirect embedding of M into $\prod_j S_j$ is an injective homomorphism $\Phi : M \hookrightarrow \prod_j S_j$ such that every projection $\pi_j \circ \Phi : M \to S_j$ is surjective.
By axiom AS4, $\phi < a \odot b \iff \phi < a$ or $\phi < b$. Therefore

$$h_\phi(a \odot b) = h_\phi(a) \odot h_\phi(b)$$

and, hence, each h_ϕ is a semilattice homomorphism.
According to AS1, $\mathbf{U}^c(\phi) \neq \emptyset$: pick $c \in \mathbf{U}^c(\phi)$, so $\phi < c$ and $h_\phi(c) = 1$. Since $\phi \neq \ominus_c$, there exists $c' \in C \setminus \mathbf{U}^c(\phi)$, so $\phi \not< c'$ and $h_\phi(c') = 0$. It follows that if $\phi \neq \ominus_c$, then h_ϕ is surjective.
We show now that the family $\{h_\phi : \phi \in A,\ \phi \neq \ominus_c\}$ separates the terms of M. Let $t \neq s$ in M. Then $t \not\leq s$ or $s \not\leq t$; say $t \not\leq s$. By axiom AS3, there exists an atom $\phi \in A$ with $\phi < t$ and $\phi \not< s$. This ϕ cannot be $\ominus_c$: by AS1 applied to all constants, $\ominus_c < c$ for every $c \in C$, hence $\ominus_c < s$ by Theorem 1 part iii. Therefore $\phi \neq \ominus_c$ and $h_\phi(t) = 1 \neq 0 = h_\phi(s)$.
Φ is a homomorphism because each coordinate h_ϕ is, Φ is also injective as it separates the terms of M and each projection $\pi_\phi \circ \Phi = h_\phi$ is surjective. Therefore, Φ is a subdirect embedding of M. □

A.3 Additional results

In this section we present a few new results on atomized semilattices necessary to support this paper.

Theorem 19. *Let $R = R^+ \cup R^-$ where R^+ is a set of atomic sentences of the form $(a \leq b)$ and R^- a set of negated atomic sentences of the form $(a \not\leq b)$. Let r, s and t be duples;*
i) if $R \Rightarrow r^+$, then either $R \cup \{r^+\}$ is inconsistent or $R^+ \Rightarrow r^+$,
ii) if $R \Rightarrow s^-$ there is at least one duple $t \in R^-$ such that $R^+ \cup \{t^-\} \Rightarrow s^-$,
where the notation $r^+ = (r_L \leq r_R)$ and $r^- = (r_L \not\leq r_R)$ has been used.

Proof. (i) Let M and N be two atomized semilattices over the same set of constants C. Considering the discriminant atoms present in the model $M + N$ it follows that $M + N \models Th_0^-(M) \cup Th_0^-(N)$ and $M + N \models Th_0^+(M) \cap Th_0^+(N)$, where $Th_0^+(M)$ and $Th_0^-(M)$ are the sets of every positive and negative duples over C, respectively, satisfied by M.
Assume $R \cup \{r^+\}$ is consistent; there is a model $M \models R^+ \cup R^- \cup \{r^+\}$. Suppose $R^+ \not\Rightarrow r^+$, there is a model $N \models R^+ \cup \{r^-\}$. Since every semilattice can be atomized, without loss of generality we can assume M and N are atomized over C, where C is the set of constants mentioned in R. Let the model $M + N$ be the model spawned by the union of the atoms of M and the atoms of N. It follows $M + N \models R^+ \cup R^- \cup \{r^-\}$ contradicting $R \Rightarrow r^+$.
(ii) Assume that for every duple $t \in R^-$, $R^+ \cup \{t^-\} \not\Rightarrow s^-$. This means that there are models N_t satisfying $\forall t(t \in R^-) \exists N_t (N_t \models R^+ \cup \{t^-\} \cup \{s^+\})$. Consider the model Q spawned by the atoms of all the models N_t. We have $Q \models R^+ \cup R^- \cup \{s^+\}$ contradicting $R^+ \cup R^- \Rightarrow s^-$. □

Theorem 20. *Let M be an atomized model and $r = (r_L, r_R)$ a duple so $M \models r^-$.*
i) Each non-redundant atom of M is non-redundant with $F(Th^+(M) \cup r^+)$.
ii) If K is an atomization of $F(Th^+(M) \cup r^+)$ without redundant atoms, any atom ϕ of K is either a non-redundant atom of M or the union of two non-redundant atoms of M, λ and ρ, such that $U^c(\phi) = U^c(\lambda) \cup U^c(\rho)$ and $\lambda < r_L$, $\lambda \not< r_R$ and $\rho < r_R$.

Proof. Theorem 11 says that a model Q for $F(Th^+(M) \cup r^+)$ can be obtained from M by full-crossing of r^+. Suppose that a non-redundant atom $\phi \in M$ becomes redundant with a set of atoms of Q. Since ϕ was not redundant in M then the set Γ of atoms of Q that makes ϕ redundant should contain at least one atom $\lambda \bigtriangledown \rho$ introduced by full-crossing (see the proof of Theorem 11), but if this is the case, then ϕ is redundant with the set $\Gamma \setminus \{\lambda \bigtriangledown \rho\} \cup \{\lambda\} \cup \{\rho\}$. If Γ contains more than one new atom of Q we can replace it by its components λ and ρ until forming a set of atoms of M that make ϕ redundant contradicting that ϕ is non-redundant in M. The situation does not change if we add redundant atoms to Q, therefore ϕ is not redundant in any atomization of Q.

(ii) follows from Theorem 11 and the uniqueness of the model without redundant atoms. The atomization of $F(Th^+(M) \cup r^+)$ without redundant atoms, K, should be the subset of non-redundant atoms of the model obtained by full-crossing of r^+ in an atomization of M without redundant atoms, so every atom of K is a non-redundant atom of M or is in the set $(M \setminus H) \cup (H \bigtriangledown B)$ where H and B are defined in the proof of Theorem 11, so any new atom of K is a union of two non-redundant atoms of M. □

Theorem 21. *Let N be an atomized model, λ an atom of N and $r = (r_L, r_R)$ a duple. Let $M = \Box_r N$. Either $\lambda \notin dis_N(r)$, and then $\lambda \in M$, or $\lambda \in dis_N(r)$ and then $\lambda \notin M$ and there is at least one and possibly various atoms in M strictly wider than λ.*

Proof. The Theorem follows from Theorem 20 and from the full-crossing mechanism given in the proof of Theorem 11. Full-crossing transforms N into a model M leaving unaltered the atoms of N except when the atom $\lambda \in dis_N(r)$, in this case λ is replaced by at least one, usually multiple, completions in M. These completions ϕ are atoms which upper constant segment $\mathbf{U^c}(\phi) = \mathbf{U^c}(\lambda) \cup \mathbf{U^c}(\rho)$ for some atom $\rho \in N$ (see Theorem 20). Since $M \models r^+$ then $dis_M(r) = \emptyset$, and since λ discriminates r, then λ cannot be in M. It follows that every completion ϕ must be different from λ, i.e. it should be strictly larger than λ. □

Theorem 22. *Let N_0 be an atomized model over a set of constants C, R^+ a set of positive duples and $r_1, r_2, ..., r_n$ an ordering of R^+. Assume R^+ is enforced in N_0 by successive full-crossings in the order given and that intermediate models $N_1, N_2, ..., N_n$ where $N_{i-1} \models r_i^-$, $N_i \models r_i^+$ and $N_n = F(Th^+(N_0) \cup R^+)$ are obtained.*

- *i) For each atom $\phi \in N_n$ there is at least one "inward chain" of atoms $\lambda_0, \lambda_1, ..., \lambda_n$ such that $\lambda_0 \in N_0$, $\phi = \lambda_n$ and $\lambda_i \in N_i$ either $\lambda_i = \lambda_{i+1}$ or λ_{i+1} is strictly larger than λ_i.*
- *ii) For each atom $\lambda \in N_0$ there is at least one "outward chain" of atoms $\lambda_0, \lambda_1, ..., \lambda_n$ such that $\lambda_0 = \lambda$ and $\lambda_i \in N_i$ with either $\lambda_i = \lambda_{i+1}$ or λ_{i+1} is strictly larger than λ_i.*
- *iii) Along any inward or outward sequence of atoms the number of times $\lambda_i \neq \lambda_{i+1}$ is at most $|U^c(\lambda_n)| - 1 < |C|$.*
- *iv) Along any inward or outward chain an atom $\lambda_i \in dis_{N_i}(r_{i+1})$ if and only if $\lambda_i \neq \lambda_{i+1}$.*
- *v) Along any inward or outward chain an atom λ_i in the chain causes model N_i to produce a false negative when tested with duple r_{i+1}^+ at most $|U^c(\lambda_n)| - |U^c(\lambda_0)| < |C|$ times.*
- *vi) Along any inward or outward chain, for each step i there is an atom $\rho_i \in N_i \cap N_{i+1}$ such that $\lambda_{i+1} = \lambda_i \bigtriangledown \rho_i$.*

Proof. Let N_{i+1} be a model obtained as a result of full-crossing of r_{i+1} in N_i. From Theorem 20, or directly from the definition of full-crossing, it follows that an atom $\lambda_{i+1} \in N_{i+1}$ is either an atom of N_i or has an upper constant segment that is the union of the upper constant segments of two atoms of

N_i, say λ_i and ρ_i, such that $\rho_i \in N_{i+1}$, $\lambda_i \notin N_{i+1}$ and $\lambda_i \in dis_{N_i}(r_{i+1})$. Write $\lambda_{i+1} = \lambda_i \bigtriangledown \rho_i$. This suffices to establish that for each atom of $\lambda_{i+1} \in N_{i+1}$ there is at least one precursor atom in $\lambda_i \in N_i$, perhaps equal to λ_{i+1}. The precursor atom may not be unique because the same λ_{i+1} may be formed by union of two different λ_i, ρ_i pairs. This proves the existence of an inward chain, i.e. (i).

To prove (ii) we can use the same reasoning. Given a $\lambda_i \in N_i$ the outward chain continues with $\lambda_{i+1} = \lambda_i$ when $\lambda_i \notin dis_{N_i}(r_{i+1})$ and in case $\lambda_i \in dis_{N_i}(r_{i+1})$ then we can choose any of the resulting atoms from the union of λ_i with any other atom ρ_i. This produces at least one, and typically many, possible outward chains.

Number (iii) follows trivially from the fact that any atom $\lambda_{i+1} = \lambda_i \bigtriangledown \rho_i$ formed by full-crossing is strictly larger than $\lambda_i \in dis_{N_i}(r_{i+1})$ and can grow at most $|U^c(\lambda_n)| - |U^c(\lambda_0)|$ times.

Number (iv) is the result of Theorem 21 applied to successive atoms in any chain. Number (v) is the same as (iv) but points out that an atom λ_i causes model N_i to produce a false negative for duple r^+_{i+1}, i.e. $N_i \models r^-_{i+1}$ only when $\lambda_i \in dis_{N_i}(r_{i+1})$. When an atom produces a false negative for r^+_{i+1} its upper constant segment acquires at least one new constant in the crossing of r_{i+1}. Atoms along any chain can fail at most $|U^c(\lambda_n)| - |U^c(\lambda_0)|$ duples from the set $r_1, r_2, ..., r_n$, irrespective of how large is n.

(vi) is a consequence of $\rho_i \in N_i$ that remains unaltered in N_{i+1} and the recipe given above for the construction of λ_{i+1}. At the stages in the chain when $\lambda_i = \lambda_{i+1}$ we can use $\rho_i = \lambda_i$ since $\lambda_{i+1} = \lambda_i = \lambda_i \bigtriangledown \lambda_i$. □

Theorem 22 proves the existence of inward and outwards chains of atoms $\lambda_0, \lambda_1, ..., \lambda_n$. Each chain ends in an atom of a model N_n obtained by crossing a set of positive duples R^+ in an order $r_1, r_2, ..., r_n$ producing a sequence of intermediate models $N_0, N_1, N_2, ..., N_n$. Each atom along the chain belong to the corresponding intermediate model: $\lambda_i \in N_i$.

B Discovery of rules in data

Let P be a set of atomic and negated atomic sentences and Q the set of its logical consequences, excluding P itself. For example, Q may represent data and P the underlying rules implicit in the data that are not directly observable.

In Appendix B.1 we prove that the freest model $F_C(Q^+)$ and the freest model $F_C(P^+)$ are atomized by the following disjoint sets of non-redundant atoms:

$$Nr(F_C(consequences)) = Nr(F_C(Q^+)) = \Phi \cup \Pi,$$
$$Nr(F_C(rules)) = Nr(F_C(P^+)) = \Phi \cup \Omega.$$

The disjoint sets of atoms Φ, Π and Ω classify the non-redundant atoms involved into three separated classes. Since P^+ implies Q^+, the model $F_C(Q^+)$ is freer than $F_C(P^+)$ and, since $Q^+ \cap P^+ = \emptyset$, some or all the sentences of P^+ are negative in $F_C(Q^+)$. The atoms in the set Π are precisely the ones discriminating duples of P^+ in $F_C(Q^+)$ and in Theorem 23 we prove that they have a cardinal bounded by that of P^+:

$$|\Pi| \leq |P^+|.$$

The atoms in Π encode that P^+ is not in Q^+ and are usually very wide (with very large upper constant segments) and, hence, barely discriminative. The atoms in the set Ω are non-redundant in $F_C(P^+)$ but redundant in $F_C(Q^+)$. Each atom $\omega \in \Omega$ is a union of at least one atom $\pi \in \Pi$ and other atoms of $F_C(Q^+)$, so they are very similar but even wider than those of Π. Furthermore, every duple discriminated by ω is discriminated by an atom of Π. If an atom in Ω discriminates a duple s, i.e. if it causes $F_C(P^+) \models s^-$ then s^+ implies (in any model) at least one duple of P^+.

The set Φ contains the useful atoms: a duple that corresponds to a negative axiom, is negative in models $F_C(P^+)$ and $F_C(Q^+)$ because it is discriminated by at least one atom in the set Φ. These atoms are common to both models.

In a realistic scenario, only a small subset of the consequences, $Q' \subset Q$, is observed (the training data). Because the upper constant segment of the atoms cannot grow without limit as a result of a sequence of crossings, the atoms mature rapidly into atoms of Φ. Some intuition about this can be obtained from Appendix C, but in essence, when an atom appears in the discriminant of a duple in Q'^+, it is replaced by one or more wider atoms. Any atom ϕ in the model is the result of an "inward chain" of length $g(\phi)$ of increasingly wider atoms (with increasing upper constant segments). Since $g(\phi)$ cannot exceed $|U^c(\phi)|$, atoms mature after a few effective crossings. Specifically, effective crossings for ϕ are only the $g(\phi)$ ones that influenced the inward chain, i.e., those in which the atom in the inward chain appears in a discriminant and is, therefore, replaced. For example, most atoms of the MNIST dataset have about 10 to 12 constants, which means that they matured in fewer than 10 to 12 effective crossings. Although most crossing have no effect on the inward chain of ϕ, if ϕ is not yet matured after $|Q'^+|$ crossings then ϕ has an expected probability of causing a false negative lower than $g(\phi)/|Q'^+|$.

The non-redundant atoms of $F_C(Q'^+)$ comprise a subset $\Phi' \subset \Phi$ and a set of immature atoms that would develop into atoms of Φ if more data were processed. Other non-redundant atoms of $F_C(Q'^+)$ include atoms of Π and also wider atoms that, if more data is processed, will end up in the discriminant of sentences of Q^+ and removed from the model without leaving an offspring, as they produce only redundant atoms when they are crossed:

$$Nr(F_C(Q'^+)) = \Phi' \cup Immature \cup Incompatible \cup \Pi.$$

If Sparse Crossing is used, a model N is obtained that depends upon the set Q'^- and is atomized by a subset of the atoms of $F_C(Q'^+)$:

$$Nr(N) \approx \Phi'' \cup Immature' \cup Incompatible' \subseteq Nr(F_C(Q'^+)),$$

where prime indicates subset, i.e. $\Phi'' \subseteq \Phi'$, etc. An atom of Π can only be discovered in the improbable case that it discriminates a duple of Q'^-. Incompatible atoms are typically large and not very discriminative, so their presence have a relative low impact in the behavior of N. The atoms of Φ'' and also the immature atoms provide an approximation to the rules P. We should take into account:

- Some atoms of Φ are rapidly discovered even when the set of examples seen, Q'^+, is a tiny fraction of the set Q^+. Since every subset of Φ produces a model that satisfies the entire set Q^+ this leads to generalization.

- There is a very large number of subsets of Φ that lead to generalizing models; in Appendix C we discussed that small subsets of the atoms of a model can approximate the model with high accuracies even in the presence of noise. Atoms of Φ not present in a model cause false positives, but since most duples discriminated by an atom of Φ are also discriminated by other atoms of Φ, small subsets of atoms can produce models that perform with low probability of false positive (but not 0).
- Different rules or aspects of the rules may be encoded by distinct atoms of Φ. Some of these atoms may not be discovered by Sparse Crossing, depending on the counterexamples provided in Q'^{-}. For instance, consider a problem of identifying patterns in binary (black and white) images. There exist atoms in Φ that encode precisely this constraint, i.e., that each pixel location contains either a black or white pixel. Since all examples, including counterexamples, adhere to this rule, these atoms are not discriminative and will not be discoverable by Sparse Crossing.
- The entire model $F_C(Q'^{+})$ is not generalizing, as it assigns negative every duple not implied by Q'^{+}, which causes false negatives. This is caused by the cumulative effect of all the incompatible and immature atoms, which cause false negatives. However, the number of atoms in $F_C(Q'^{+})$ is typically immense, and this loss of generalization is not observed in practice. Models with very different number of atoms tend to produce similarly generalizing models (see figure S6).
- There are other nuances that should be taken into account. For example, observable sentences in Q' typically belong to a localized subset of the space of possible duples. In the case of binary (black and white) images, all observable terms have as many constants as pixels. This can have the effect of introducing implicit rules in the data, see Theorem 24.
- There are usually many symmetries given by permutations of the constants that interchange atoms of Φ leaving this set invariant. This property can potentially lead to reasonable guesses of Φ from a subset of it.

B.1 Learning the causal theory

Theorem 23. *Let P and Q be two sets of signed duples over a set of constants C such that Q is the set of logical consequences of P minus the set P:*

i) The atomization without redundant atoms of $F_C(Q^+)$ consists of non-redundant atoms of $F_C(P^+)$ and an additional set of atoms of cardinal equal or lower than that of $|P^+|$.

ii) Each non-redundant atom of $F_C(Q^+)$ is either a non-redundant atom of $F_C(P^+)$ or is the unique discriminant of one duple of P^+.

iii) If a non-redundant atom of $F_C(Q^+)$ that is the unique discriminant of a duple $s^+ \in P^+$ discriminates a duple $r = (r_L, r_R)$ then the component constants of the term r_R are component constants of s_R.

iv) For each non-redundant atom ω of $F_C(P^+)$ that is not a non-redundant atom of $F_C(Q^+)$ there is a pining duple $u = (c, T_\omega)$ of ω such that there is a $r^+ \in P$ for which, $u^+ \Rightarrow r^+$.

Proof. By definition $P \Rightarrow Q$ and $Q \cap P = \emptyset$. Since $P \Rightarrow Q$ all the positive duples in Q are positive duples of any model of P and, hence, $F_C(Q^+)$ is freer than $F_C(P^+)$ or equal to $F_C(P^+)$. Choose a non-redundant atom of $F_C(Q^+)$ and consider the model K spawned by the chosen atom plus the atoms of $F_C(P^+)$.
Suppose $K = F_C(P^+)$. If K is equal to $F_C(P^+)$ then the chosen atom, say ϕ, is, according to Theorem 5, either redundant with $F_C(P^+)$ or is a non-redundant atom of $F_C(P^+)$. We are going to show that ϕ cannot be redundant with $F_C(P^+)$. Suppose it is redundant; ϕ is a union of atoms of $F_C(P^+)$. Since $F_C(Q^+)$ is freer than $F_C(P^+)$ Theorem 17 tells us that each atom of $F_C(P^+)$ is either an atom of $F_C(Q^+)$ or a union of atoms of $F_C(Q^+)$. It immediately follows that ϕ is a union of atoms of $F_C(Q^+)$, which contradicts the assumption that we chose a non-redundant atom of $F_C(Q^+)$. Therefore, if K is equal to $F_C(P^+)$ then ϕ is a non-redundant atom of $F_C(P^+)$.
Suppose now that $K \neq F_C(P^+)$. In this case we are going to refer to the chosen atom as π. Since K has all the atoms of $F_C(P^+)$, both models can differ only if there is at least one duple r discriminated by π such that $F_C(P^+) \models r^+$ and $F_C(K) \models r^-$. Since π is an atom of $F_C(Q^+)$ then

$F_C(Q^+) \models r^-$. Because r^+ is modeled by the freest model of P then r^+ is a logical consequence of P^+ and, by definition of Q, either $r^+ \in Q^+$ or $r^+ \in P^+$. Since $F_C(Q^+) \models r^-$, it follows that r^+ is a duple of P^+.
So far we have shown that each non-redundant atom of $F_C(Q^+)$ is either like ϕ or like π. Atoms like ϕ are also non-redundant atoms of $F_C(P^+)$ while atoms like π are never atoms of $F_C(P^+)$ and discriminate each at least one duple of P^+.
As in the proof of Theorem 15, for each r discriminated by π there is a constant $c \in U^c(\pi)$ such that the pinning duple $s = (c, T_\pi)$ satisfies $s^- \Rightarrow r^-$. Each atom π discriminates at least one duple of P^+, so let r be one of such duples. The equivalent $r^+ \Rightarrow s^+$ implies that s^+ is either a consequence of P^+ or in P^+. Since $F_C(Q^+) \models s^-$ we can discard that s^+ is a consequence of P^+ and, therefore, s^+ should be a duple of P^+. Since the pinning duples of different atoms are all different (see Theorem 15 (iii)) then each atom π can be mapped to a different duple of P^+ and, hence, $F_C(Q^+)$ cannot have more atoms like π than duples are in P^+. We have proven (i) and (ii).
iii) We have just seen that a non-redundant atom of $F_C(Q^+)$ that is the unique discriminant of a duple $s^+ \in P^+$ is an atom π_0 of type π and $s = (c, T_{\pi_0})$ is a pinning duple of π_0. If π_0 discriminates a duple $r = (r_L, r_R)$ then $\pi_0 < r_L$ and $r_R < T_{\pi_0} = s_R$ is true in any model over C, i.e. the component constants of the term r_R are component constants of s_R.
iv) Let ω be a non-redundant atom of $F_C(P^+)$ that is not a non-redundant atom of $F_C(Q^+)$. According to Theorem 6, if ω is non-redundant in $F_C(P^+)$ there is at least one pinning duple $u = (c, T_\omega)$ discriminated only by ω in $F_C(P^+)$. Because $F_C(Q^+)$ is freer than $F_C(P^+)$ then $F_C(Q^+) \models u^-$. The non-redundant atoms of $F_C(Q^+)$ discriminating u^- cannot be of type ϕ because these atoms are also atoms of $F_C(P^+)$ and that would contradict that u is discriminated only by ω in $F_C(P^+)$. Therefore, the atoms of $F_C(Q^+)$ that discriminate u are of type π.
Since $F_C(Q^+)$ is freer than $F_C(P^+)$ then ω is a union of atoms of $F_C(Q^+)$. Among these atoms at least one is of type π, otherwise ω would be redundant in $F_C(P^+)$ and it is not. In fact, at least one of these atoms of type π should discriminate u simply because if atom $\alpha \bigtriangledown \beta$ discriminates u then either α or β discriminates u. Let $\pi_0 \in F_C(Q^+)$ be one of such atoms of type π that discriminates u, i.e. $\pi_0 < c$ and $\pi_0 \not< T_\omega$. Since ω is wider than π_0, $T_\omega < T_{\pi_0}$. It follows $(c \not\leq T_{\pi_0}) \Rightarrow u^-$. We showed above that the pinning duples of the atoms of type π are elements of the set P^+. Therefore, $u \in P^+$ and $u^+ \Rightarrow (c \leq T_{\pi_0})$. □

Often the duples that can be observed lie in a region, $W \subseteq F_C(\emptyset) \times F_C(\emptyset)$, of the space of duples. For example, if we are learning images, the duples observed may all have a term at the right-hand side that correspond to an image; a term formed as a summation of as many constants as pixels. Terms with other number of component constants may not be accessible for training. The following theorem clarifies this situation:

Theorem 24. *Let P and Q be two sets of signed duples over a set of constants C such that Q is the set of logical consequences of P minus the set P. Let W by the set of "observable duples":*

- *i) The non-redundant atoms of $F_C(P^+)$ that are non-redundant atoms of $F_C(Q^+ \cap W)$ are also non-redundant atoms of $F_C(Q^+)$.*
- *ii) The non-redundant atoms of $F_C(P^+)$ that are non-redundant atoms of $F_C(Q^+)$ are atoms of $F_C(Q^+ \cap W)$, redundant or non-redundant.*
- *iii) A non-redundant atom η of $F_C(P^+)$ that is a non-redundant atom of $F_C(Q^+)$ but a redundant atom of $F_C(Q^+ \cap W)$ exists only if there is a non-redundant atom δ of $F_C(Q^+ \cap W)$ that is narrower that η and discriminated by at least one duple of $Q^+ \cap \overline{W}$ and by no duple of $Q^+ \cap W$.*

Proof. Consider the sequence of models:

$$F_C(Q^+ \cap W) \rightarrow F_C(Q^+) \rightarrow F_C(P^+) = F_C(P^+ \cup (Q^+ \cap \overline{W}))$$

were each $\rightarrow$ represent a constant-preserving homomorphism. The last equality follows easily as $P^+ \Rightarrow Q^+$ and then the freest model of $P^+ \cup (Q^+ \cap \overline{W})$ is the same of that of P^+ alone. It is clear that the models in the sequence are increasingly less free as we move to the right; $F_C(Q^+ \cap W)$ is obviously freer or equal to $F_C(Q^+)$ and $F_C(Q^+)$ is freer than $F_C(P^+)$ because every duple in Q^+

is a logical consequence of P^+.
Let's refer to the pair of models $F_C(Q^+ \cap W)$ and $F_C(P^+ \cup (Q^+ \cap \overline{W}))$ as the "external pair" and to the pair of models $F_C(Q^+)$ and $F_C(P^+)$ as the "internal pair".
For the internal pair, the first model, i.e. $F_C(Q^+)$ contains the logical consequences (but not the premises) of the second, $F_C(P^+)$, with an empty intersection $Q^+ \cap P^+ = \emptyset$. Just like the internal pair, the external pair behaves in the same manner; $Q^+ \cap W$ is the set of logical consequences of $P^+ \cup (Q^+ \cap \overline{W})$ and both sets have empty intersection.
In the proof of Theorem 23 we showed that the non-redundant atoms of the (freest model of) the consequences where, either non-redundant atoms of the causes, we called them type ϕ atoms, or they were external to the model of the causes and we refereed to that atoms as of type π. Also, we used ω for the non-redundant atoms of the causes that were redundant atoms of the consequences; let's refer to them as atoms of type ω.
i) The atoms of type ϕ for the external pair are non-redundant atoms of $F_C(P^+ \cup (Q^+ \cap \overline{W})$, which is equal to $F_C(P^+)$. Since a non-redundant atom of the first model of the sequence that is also an atom of the last model of the sequence is a non-redundant atom of the four models of the sequence, it is clear that the atoms of type ϕ for the external pair are also atoms of type ϕ for the internal pair. The atoms of type ϕ we are discovering using duples of W are atoms of type ϕ that we would discover using all the duples.
ii) It follows from the fact that $F_C(Q^+ \cap W)$ is freer than $F_C(Q^+)$. Notice that the type ϕ atoms of the internal pair that are not type ϕ atoms of the external pair are redundant atoms of $F_C(Q^+ \cap W)$ that become non-redundant when crossing some duple of $Q^+ \cap \overline{W}$, since they are non-redundant for $F_C(Q^+)$.
iii) It is natural to wonder which type ϕ atoms of the internal pair are not discovered when using only duples of W. Since η is redundant in $F_C(Q^+ \cap W)$ it must be equal to a union of non-redundant atoms of $F_C(Q^+ \cap W)$. As η becomes non-redundant when crossing the duples in $Q^+ \cap \overline{W}$ there must be some non-redundant atom δ of $F_C(Q^+ \cap W)$ narrower than η, i.e. $U_C(\delta) \subseteq U_C(\eta)$, that is discriminated away by some duple of $Q^+ \cap \overline{W}$. Notice that δ is of type π for the external pair.
□

Example. Consider the task of, given a black and white $n \times n$ image, distinguishing whether or not it has at least one black vertical bar. We may have an underlying theory

$$P^+ = \{(v \leq bar_1), (v \leq bar_2), ..., (v \leq bar_n)\}$$

with $2n^2 + 1$ constants, one for each color of the n^2 pixels plus one constant for the vertical bar v. If the observable duples belong to well-formed images, then W consists of the set of all the terms with n^2 constants, one for each pixel. The constant are $C = \{v, b_{ij}, w_{ij} : i, j \in \{1, ..., n\}\}$ and the atomization of the freest model is

$$F_C(\emptyset) = \{\phi_v, \phi_{b_{ij}}, \phi_{w_{ij}} : i, j \in \{1, ..., n\}\}$$

where the b_{ij} are the constants for the black color pixels and w_{ij} the constants for the white color pixels. We then have:

$$F_C(P^+) = \{\phi_{b_{ij}}, \phi_{w_{ij}}, \phi_{v,b_{\sigma(1)1},b_{\sigma(2)2},...,b_{\sigma(n)n}} : i, j \in \{1, ..., n\} \;\; \sigma : \{1, ..., n\} \to \{1, ..., n\}\}$$

where σ runs along the set of all functions from $\{1, ..., n\}$ to $\{1, ..., n\}$. The atomization has $2n^2$ atoms with one constant in its upper constant segment plus n^n atoms with n constants in their upper constant segments, one constant for a black pixel selected for each of the n columns. In total $2n^2 + n^n$ non–redundant atoms.

$$F_C(Q^+) = \{\phi_{b_{ij}}, \phi_{w_{ij}}, \phi_{v,b_{\sigma(1)1},b_{\sigma(2)2},...,b_{\sigma(n)n}} : i, j \in \{1, ..., n\} \;\; \sigma : \{1, ..., n\} \to \{1, ..., n\}\} \cup$$
$$\cup\{\phi_{C-\{b_{1j},b_{2j},...,b_{nj}\}}, : j \in \{1, ..., n\}\}.$$

Every non-redundant atom of $F_C(P^+)$ is discovered by $F_C(Q^+)$. The n atoms $\phi_{C-\{b_{1j},b_{2j},...,b_{nj}\}}$ are type π atoms in the proof of Theorem 23 (i) and each of them have a duple of P as pining duple. We will see later that:

$$F_C(Q^+ \cap W) = \{\phi_{b_{ij}}, \phi_{w_{ij}}, \phi_{v,b_{\sigma(1)1},b_{\sigma(2)2},...,b_{\sigma(n)n}} : i, j \in \{1, ..., n\} \;\; \sigma : \{1, ..., n\} \to \{1, ..., n\}\}$$
$$\cup\{\phi_{v,b_{ij},w_{ij}} : i, j \in \{1, ..., n\}\}.$$

Again, every non-redundant atom of $F_C(P^+)$ is discovered by $F_C(Q^+ \cap W)$. The n^2 atoms $\phi_{v,b_{ij},w_{ij}}$ actually characterize the set W. Notice that the π atoms $\phi_{C-\{b_{1j},b_{2j},\ldots,b_{nj}\}}$ are equal to unions of atoms of the form $\phi_{v,b_{ij},w_{ij}}$ and $\phi_{w_{ij}}$, so they are all redundant with $F_C(Q^+ \cap W)$. From Theorem 24 (iv) we can deduce that there is no non-redundant atom of $F_C(P^+)$ that is non-redundant in $F_C(Q^+)$ and redundant in $F_C(Q^+ \cap W)$; an atom δ narrower to $\phi_{b,b_{\sigma(1)1},b_{\sigma(2)2},\ldots,b_{\sigma(n)n}}$ is necessarily missing some black pixel constant in some of the n columns and it is, therefore, discriminated by some duple $Q^+ \cap W$, i.e. there is a duple with a term of a well-formed image that has δ in its discriminant. Therefore, every type ϕ atom of $F_C(Q^+)$ is discovered by $F_C(Q^+ \cap W)$.

Notice that for any pattern recognition problem of this kind, with a property v shared by a set of binary images, we can say the same as for the vertical bar. The atoms of v that discriminate images without the property v cannot have the black and white constant for the same pixel. Therefore, there is no non-redundant atom of $F_C(P^+)$ that is a non-redundant in $F_C(Q^+)$ and redundant in $F_C(Q^+ \cap W)$. Hence, $F_C(Q^+ \cap W)$ discovers every non-redundant atom of $F_C(P^+)$ discovered by $F_C(Q^+)$.

C Generalizing Subsets of the Freest Model

We first give a summary of the results of this supplement. We consider the freest model $M = F_C(R^+)$ of an ideal set of positive duples containing every valid sample of a task, for example, a classification problem. Since R^+ contains every valid sample, M must perform with 0 error. Given an ordering $R^+ = \{s_1^+, s_2^+, s_3^+, ..., s_J^+\}$ and applying full-crossing, we obtain the sequence of models $F_0, F_1, ..., F_j, ..., F_J$, with $F_j = F_C(s_1^+, s_2^+, ..., s_j^+)$ for $0 \leq j \leq J = |R^+|$ with $F_0 = F_C(\emptyset)$ and $F_J = M$, which is independent of the order chosen. We are interested in calculating how well a subset of atoms of the intermediate model F_j approaches M.

In Appendix C.1 we show that, in the task of approaching M with a subset of F_j, after crossing j positive duples, the expected value for the probability of false negative of a subset of Z atoms of F_j is given by

$$\overline{\text{PFN}}(\phi_1, \ldots, \phi_Z) \approx \sum_{i=1}^{Z} \min\left(\frac{1}{h(\phi_i)+1}, \frac{g(\phi_i)+1}{j+1}\right),$$

where $g(\phi_i)$ is the length of the inner chain of atom ϕ_i, i.e. the number of atom unions that took the formation of atom ϕ_i, and $h(\phi_i)$ is the number of positive duples crossed after the final atom union that produced ϕ_i. The expected PFN decreases with j so, for a sufficiently large j it becomes as small as desired. Eventually, the expected PFN is dominated by the length of the tails $h(\phi_i)$;

$$\overline{\text{PFN}}(\phi_1, \ldots, \phi_Z) \approx \sum_{i=1}^{Z} \frac{1}{h(\phi_i)+1} \approx \sum_{i=1}^{Z} \frac{1}{h(\phi_i)},$$

which we use in main text. Since the tails grow linearly with j the collective PFN still decreases with j and grows linearly with the number of atoms in the set, Z.

To find out if our subset is generalizing we must also consider the probability of false positive. A duple that must be negative gives a false positive if every atom in our subset fails to discriminate the duple. In this case

$$\overline{\text{PFP}}(\phi_1, \ldots, \phi_Z) = \prod_{i=1}^{Z} \overline{\text{PFP}}(\phi_i),$$

which we have considered in the main text. This expression assumes the $\overline{\text{PFP}}(\phi_i)$ of individual atoms are approximately independent of each other. The assumption of independence may lead to a safe overestimation of $\overline{\text{PFN}}$ but to an underestimation of the $\overline{\text{PFP}}$. However, because every F_j is freer than M, there must be atoms in F_j that discriminate every duple that is negative in M. With a large enough subset, and even if there are correlations, we must get a collective $\overline{\text{PFP}}$ as small as desired, even equal to 0.

To minimize the $\overline{\text{PFP}}$ we have to select the atoms for our subset as those having a low and uncorrelated individual $\overline{\text{PFP}}(\phi_i)$, which we can do by computing $\overline{\text{PFP}}(\phi_i)$ using the negative duples of the training set. In addition, we should also make sure to select a subset that, collectively, discriminates all the negative duples of the training set. Since correlated atoms discriminate similar duples, the smaller the resulting subset of atoms the less correlation we must have.

Since the upper constant segment of an atom is limited by the number of constants, atoms cannot "grow" without limit. Every atom present in the hypothetical model M is the result of a finite number of atom unions and after that, the atom "matures" and remains unchanged. Notice that an atom is removed from an intermediate model when it causes the first false negative in the sequence $s_1^+, s_2^+, s_3^+, ..., s_J^+$, then the atoms grows, but this cannot happen many times. In fact, atoms obtained in classification problems have just a few constants in its upper constant segment: 5 to 50 constants, with most atoms around 12 to 15 constants, are typical values observed, so maturity is typically reached after $g(\phi_i) < 15$ crossings.

We are in an advantageous situation where the $\overline{\text{PFP}}$ decreases geometrically with Z while the $\overline{\text{PFN}}$ increases linearly with Z but decreases inversely proportional to j. We then expect subsets of atoms of the freest model F_j with low and uncorrelated individual $\overline{\text{PFP}}$ to do a good job approaching M, with an error as small as desired as j increases, which is, indeed, observed experimentally in a wide range of problems.

Finally, we consider the test errors in practice. Sparse Crossing is the practical algorithm we designed to obtain the atoms that give low test PFN and low test PFP. In Appendix C.2 we give numerical results of PFN and PFP when using Sparse Crossing.

C.1 Derivation of the Probability of False Negative

According to Theorem 11 we can calculate the freest model $F_C(R^+)$ using the full-crossing of the duples in R^+ in any order. Suppose we want the algebra to learn a model $M = F_C(R^+)$. We can choose an ordering for the set of its positive duples $R^+ = \{s_1^+, s_2^+, s_3^+, ..., s_J^+\}$ and apply full-crossing. Consider the sequence of models $F_0, F_1, ..., F_j, ..., F_J$, with $F_j = F_C(s_1^+, s_2^+, ..., s_j^+)$ for $0 \leq j \leq J = |R^+|$ with $F_0 = F_C(\emptyset)$ and $F_J = M$. We want to derive the probability of false negative, ω, incurred by one or many atoms of model F_j in the task of approaching M.

The more widely used False Positive Ratio (FPR) becomes approximately equal to the probability of false negative (false negatives divided by true positives) when it is small, i.e. when the number of false negatives is much smaller than the number of true positives.

For a typical pattern recognition task, R^+ would correspond with the set of all valid positive samples that could exist, so M gives no error in the task. The positive duples of the training set would then correspond to the initial samples of an ordering given to R^+.

When s_j is presented to F_{j-1} it produces a false negative if $F_{j-1} \models s_j^-$, which implies that there is one or many atoms of F_{j-1} in the discriminant $dis_{F_{j-1}}(s_j^-)$. In the crossing of duple s_j each atom in the discriminant produces wider atoms (perhaps redundant) and then is removed from the model. This occurs if and only if s_j is a false negative of F_{j-1}.

Full-crossing is a process of elimination of atoms and creation of new atoms as unions, $\phi \bigtriangledown \psi$, of two existing atoms. Suppose an atom $\phi \bigtriangledown \psi$ is formed when crossing duple s_k, i.e. at "inception time" k, and assume that this atom has survived until time j with $k < j << J$, i.e. assume $\phi \bigtriangledown \psi$ is an atom of F_j. If $\phi \bigtriangledown \psi$ belongs to F_j then $\phi \bigtriangledown \psi$ is compatible with the $j-k$ duples $s_{k+1}^+, s_{k+2}^+, ..., s_j^+$ or, in other words, the atom is in the discriminant of none of these $j-k$ duples. Knowing that the atom has been compatible with $j-k$ positive training duples can be used to calculate a Bayesian estimation of the cumulative distribution of the probability of false negative $\omega(\phi \bigtriangledown \psi)$; this calculation is carried out in Section C.1.1. The cumulative distribution for the probability of false negative of atom $\phi \bigtriangledown \psi$ is:

$$P(\omega(\phi \bigtriangledown \psi) \geq \delta) \; = \; (1-\delta)^{j-k(\phi \bigtriangledown \psi)+1}$$

where j is the moment at which we measure the probability of false negative, $k(\phi \bigtriangledown \psi)$ is the inception time of the atom and $P(\omega(\phi \bigtriangledown \psi) \geq \delta)$ is the probability for the probability of false negative of $\phi \bigtriangledown \psi$ to be larger than a value δ.

Without loss of generality assume that atom ϕ is in the discriminant of duple $s_{k(\phi \bigtriangledown \psi)}$ so $\phi \bigtriangledown \psi$ is created from atom ϕ during the crossing of $s_{k(\phi \bigtriangledown \psi)}$. The cumulative distribution for atom ϕ can be approached as $P(\omega(\phi) \geq \delta) \; \approx \; (1-\delta)^{k(\phi \bigtriangledown \psi)-k(\phi)}$ and then:

$$P(\omega(\phi) \geq \delta) P(\omega(\phi \bigtriangledown \psi) \geq \delta) \; \approx \; (1-\delta)^{j-k(\phi)+1}.$$

see Section C.1.1 for details. We can apply the same formula along any inward chain $\varphi_0, \varphi_1, ..., \varphi_g$ leading to atom $\varphi_g = \phi \bigtriangledown \psi$ that exists at time j (see Theorem 22), to get the product:

$$\Pi_{n=0}^{g} P(\omega(\varphi_n) \geq \delta) \; \approx \; (1-\delta)^{j+1}.$$

where $k(\varphi_0) = 0$ has been used. As more positive duples are presented to the algebra the atoms become wider and tend to produce fewer false negatives. Assuming the last atom φ_g in the chain does not produce false negatives with significantly higher probability that the atoms along the chain, we get:

$$P(\omega(\varphi_g) \geq \delta) \; \leq \; (1-\delta)^{\frac{j+1}{g+1}}.$$

This bound has a straightforward interpretation; atoms in the chain cause false negatives at times $k(\varphi_0), k(\varphi_1), ..., k(\varphi_g)$. If the atoms get increasingly better at producing false negatives, as we progress along the chain the intervals between false negatives $k(\varphi_1) - k(\varphi_0), k(\varphi_2) - k(\varphi_1), ..., k(\varphi_g) - k(\varphi_{g-1}), j - k(\varphi_g)$ would tend to be increasingly larger, so at time j we expect to have $j - k(\varphi_{g-1}) \geq \frac{j}{g+1}$. Since the atoms cannot grow without limit, for a sufficiently large amount

of training, i.e. a sufficiently large j, the value of g becomes much smaller than j and $P(\omega(\varphi_g) \geq \delta)$ becomes small. For a large enough j, an atom of F_j produces as few false negatives as desired or no false negatives at all if the atom has matured into an atom of M.

It is clear that g can be at most equal to j and usually is much smaller than j. Each of the chains corresponds to an atom formed during a crossing and is wider than the previous atom, i.e., $U^c(\varphi_n) \subseteq U^c(\varphi_{n+1})$. Since atoms grow at each of the g links, the value of g can be bounded by how much an atom can grow:

$$g \leq |U^c(\varphi_g)| - |U^c(\varphi_0)| < |C|.$$

The number of crossings, g, can be as large as $|U^c(\varphi_g)| - |U^c(\varphi_0)|$ but also as small as 0. After a crossing, an atom of the discriminant produces unions with other atoms of any size, so its upper segment $U^c(\varphi)$ can grow in a single step by one or many constants.

In conclusion, after a sufficiently large number of positive training examples j, we can expect an atom of F_j to produce very few or no false negatives. We are assuming that the distribution of the test and training sets are equal and that we do no have positive duples that are false in the training set; that would be the case if we had mislabels, for example.

In Section C.1.1 we study various approximations to $P(\omega(\varphi_g) \geq \delta)$ for an atom φ_g present in F_j, and conclude that

$$P(\omega(\varphi_g) \geq \delta) = (1-\delta)^{a(\phi)} \quad \text{with} \quad a(\phi) = \max\left(h_g + 1, \frac{j+1}{g+1}\right)$$

where $h_g = j - k(\varphi_g)$ is the length of the last interval, approaches the confidence quite well. For this approximation, the expected value for the probability of false negative is:

$$\overline{\omega}(\varphi_g) = \min\left(\frac{g+1}{j+2}, \frac{1}{h_g+2}\right).$$

Since $a(\phi)$ is large, we can approach the expected value of the individual probability of false negative with $\overline{\omega}(\varphi_g) \approx \frac{1}{a(\phi)}$ which has a quite straightforward interpretation.

In Section C.1.2 we derive the probability of false negative produced by a set of atoms. Let W be the collective probability of false negative $\omega(\phi_1, \phi_2, ..., \phi_Z)$, for a set of Z atoms each with a probability of false negative given by $\omega(\phi_i)$. Since the model produces a false negative if any of the Z atoms produce a false negative we get:

$$\omega(\phi_1, ...\phi_Z) = 1 - \prod_{i=1}^{Z}(1 - \omega(\phi_i)).$$

We are assuming here that atoms produce false negatives in an independent manner, which is false but safe, as it overestimates the probability of false negative.

To get a small probability of false negative for Z atoms the individual probability of false negative for each atom should be quite small, so we can do a first order approximation:

$$\overline{\omega}(\phi_1, ...\phi_Z) \approx \sum_{i=1}^{Z} \overline{\omega}(\phi_i) \approx \sum_{i=1}^{Z} \frac{1}{a(\phi_i)} = \sum_{i=1}^{Z} \min\left(\frac{1}{h(\phi_i)+1}, \frac{g(\phi_i)+1}{j+1}\right). \tag{S1}$$

Since the values of $a(\phi_i)$ are known, we can compute the expected probability of false negative. Notice that this calculation can produce values larger than 1. In this case we expect a probability of false negative equal to 1, but we can still use the formula to measure how much learning is taking place even in the regime when learning cannot be measured by computing the probability of false negative experimentally with a validation dataset.

Given a value Δ for the probability of false negative of Z atoms, in Section C.1.2 we calculate $P(\omega(\phi_1, ...\phi_Z) \geq \Delta)$, and show that it is a function that transitions very abruptly from high to low values. This means that there is a moment from which we suddenly have high confidence our collective probability of false negative is better than Δ. We give an expression for the transition point from which we can estimate how much training is still needed to achieve with confidence a desired probability of false negative value.

Although the calculation becomes a bit technical, we can give an intuition of what happens, as follows. If we want the probability of false negative of a model of Z atoms to be bounded by Δ, i.e. $\overline{\omega}(\phi_1, ...\phi_Z) < \Delta$, it suffices with having each atom ϕ_i with an individual probability of false negative of $\omega(\phi_i) < \frac{\Delta}{Z}$:

$$P\left(\omega(\phi_i) \geq \frac{\Delta}{Z}\right) \leq \left(1 - \frac{\Delta}{Z}\right)^{a(\phi_i)}.$$

If we want $P(\omega(\phi_i) \geq \frac{\Delta}{Z})$ to be smaller than $\frac{1}{2}$, it is enough with requiring

$$\frac{Z}{a(\phi_i)\ \Delta} < 1,$$

that is valid for small values ofΔ. i.e. when $\Delta \approx -\ln(1-\Delta)$ is a valid approximation. We also show that the better approximation

$$\frac{1}{\Delta}\sum_{i=1}^{Z}\frac{1}{a(\phi_i)} < 1$$

also works. To derive from here an upper bound for the necessary number of training examples we can use $a(\phi_i) \leq \frac{j+1}{g+1}$ which gives:

$$j \geq \frac{\sum_{i=1}^{Z}(g(\phi_i)+1)}{\Delta},$$

and, since the values $g(\phi_i)$ are known, we can use it to estimate how much training is left to achieve a probability of false negative better than Δ.

We can use the fact that $g(\phi_i)+1$ is always smaller than the number of constants $|C|$ (each time an atom causes a false negative its upper segment grows by at least one constant), to get the bound:

$$j \geq \frac{Z\,|C|}{\Delta},$$

that is simple but significantly overestimates the amount of training needed. It implies that any model over C is approached with better probability of false negative than Δ by a set of Z atoms obtained after crossing fewer than $\frac{Z\,|C|}{\Delta}$ positive duples. The bound works even for $Z = 0$, which gives a probability of false negative equal to 0 at the expense of obtaining a probability of false positive equal to 1.

C.1.1 Probability of False Negative of a single atom

The false negatives produced by an individual atom can be illustrated with a game of biased coins. We toss j times and get g tails with $g << j$. This is a well known game but with an added difficulty: each time we get tails we have the coin replaced with a new coin that is more biased towards heads than the previous one. If we obtain g tails $g+1$ coins are played, the coins $\varphi_0, \varphi_1, ..., \varphi_g$. We are interested in determining the probability distribution for the coin φ_g at the end of the game.

We can make a first approximation using the fact that the coin we get at the end of the game, the final coin φ_g, should be more biased towards heads than a coin from which we expect g tails in j tosses. If $\omega(\varphi_g)$ is the probability for the final coin to produce tails, a lower bound for the confidence, i.e. the probability of having $\omega(\varphi_g) \geq \delta$ is:

$$P(\omega(\varphi_g) \geq \delta) \leq \frac{\binom{j}{g}\int_{\delta}^{1} t^g(1-t)^{j-g}dt}{\binom{j}{g}\int_{0}^{1} t^g(1-t)^{j-g}dt} = 1 - I_{\delta}(g+1, j-g+1)$$

where $I_{\delta}(g+1, j-g+1)$ is the regularized incomplete beta function. This expression corresponds to the Bayesian estimation of the conditional probability of having $\omega(\varphi_g) \geq \delta$ when g tails and $j-g$ heads are produced, assuming a flat prior distribution for $\omega(\varphi_g)$. This approximation uses all the information available; however, if the final coin is significantly more biased towards heads than the previous coins it overestimates the probability of obtaining tails (i.e. the probability of getting a false negative).

In the case where the final coin is significantly more biased than the previous coins a simpler approximation can yield better results. We can count how many times the final coin φ_g, produces

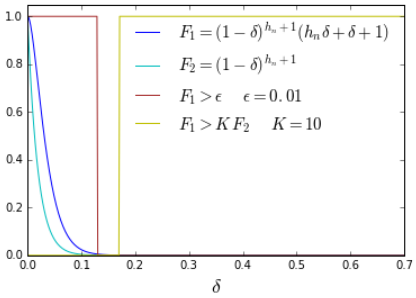


Figure S1: **Approximation to the confidence** $P(\omega(\varphi_n) \geq \delta)$**.** $F_1 = (1-\delta)^{\frac{j+1}{g+1}} \Pi_{n=0}^{g-1}(h_n\delta + \delta + 1)^{\frac{1}{g+1}}$ in dark blue and its approximation $F_2 = (1-\delta)^{h_n+1}$ in cyan color. The region of interest is located to the left of the red line while the region of excessive underestimation starts at the yellow line.

heads. We never see the final coin producing tails; we only see the final coin producing a final subsequence of h_g heads:

$$P(\omega(\varphi_g) \geq \delta) \;=\; (1-\delta)^{h_g+1},$$

that corresponds with $1 - I_\delta(1, h_g + 1)$, the formula above for h_g heads and 0 tails. Note that when $h_g = 0$ we get $P(\omega(\varphi_g) \geq \delta) \;=\; (1-\delta)$ which reflects the lack of prior information regarding the distribution of $\omega(\varphi_g)$ we have assumed. This approximation can be good under some circumstances but bad in others, particularly when the final coin has been obtained close to the end of the game and it has been tossed only a few times.

We can improve our calculations by using $P(\omega(\varphi_g) \geq \delta) \;=\; (1-\delta)^{h_g+1}$ for the final coin and:

$$P(\omega(\varphi_n) \geq \delta) \;=\; (1-\delta)^{h_n+1}(h_n\delta + \delta + 1),$$

for the intermediate coins, which corresponds with $1 - I_\delta(2, h_n + 1)$, the formula above for h_n heads and 1 tail at the end. Multiplying the $g+1$ confidence values:

$$\Pi_{n=0}^{g} P(\omega(\varphi_n) \geq \delta) \;=\; (1-\delta)^{h_g+1} \Pi_{n=0}^{g-1}(1-\delta)^{h_n+1}(h_n\delta + \delta + 1) =$$
$$= \; (1-\delta)^{1+g+\Sigma_{n=0}^{g} h_n} \Pi_{n=0}^{g-1}(h_n\delta + \delta + 1) = \; (1-\delta)^{j+1} \Pi_{n=0}^{g-1}(h_n\delta + \delta + 1)$$

where we have used $g + \Sigma_{n=0}^{g} h_n = j$, and taking into account that the coins get increasingly more biased:

$$P(\omega(\varphi_{n+1}) \geq \delta) \;\leq P(\omega(\varphi_n) \geq \delta)$$

which implies $\Pi_{n=0}^{g} P(\omega(\varphi_n) \geq \delta) \geq P(\omega(\varphi_g) \geq \delta)^{g+1}$, we get an upper bound for $P(\omega(\varphi_g) \geq \delta)$:

$$P(\omega(\varphi_g) \geq \delta) \;\leq\; (1-\delta)^{\frac{j+1}{g+1}} \Pi_{n=0}^{g-1}(h_n\delta + \delta + 1)^{\frac{1}{g+1}}.$$

Conveniently, we can ignore the multiplying factor $\Pi_{n=0}^{g-1}(h_n\delta + \delta + 1)^{\frac{1}{g+1}}$ because it is important only for values of δ we are not interested in. To be more specific, by omitting the multiplicative factor $(h_n\delta + \delta + 1)$ the probability $P(\omega(\varphi_n) \geq \delta)$ becomes underestimated by a factor of K when $(1-\delta)^{h_n+1}(h_n\delta + \delta + 1) > K\,(1-\delta)^{h_n+1}$ and, since we are interested in the region where $\omega(\varphi_n)$ is most likely higher than δ and not in the region where we can assure that (i.e. $\omega(\varphi_n) \geq \delta$) with unnecessarily high confidence, we need a good estimation in the region $(1-\delta)^{h_n+1}(h_n\delta+\delta+1) > \epsilon$ for a sufficient but modest confidence value of, say, $\epsilon > 0.005$. It should be clear that with this inequality we are limiting the confidence for the estimation of the probability of false negative and not the actual value $\omega(\varphi_n)$. The two inequalities can be rewritten as:

$$(h_n + 1)\delta > K - 1$$
$$\ln((h_n + 1)\delta + 1) - (h_n + 1)\delta > \ln(\epsilon)$$

where we have approached $\ln(1-\delta) \approx -\delta$. Both inequalities hold simultaneously for some domain of δ if and only if $\ln(K) - K + 1 > \ln(\epsilon)$, an inequality that does not depend upon h_n or δ. Therefore, provided that the inequalities do not hold simultaneously, i.e. when:

$$\ln(K) - K + 1 < \ln(\epsilon),$$

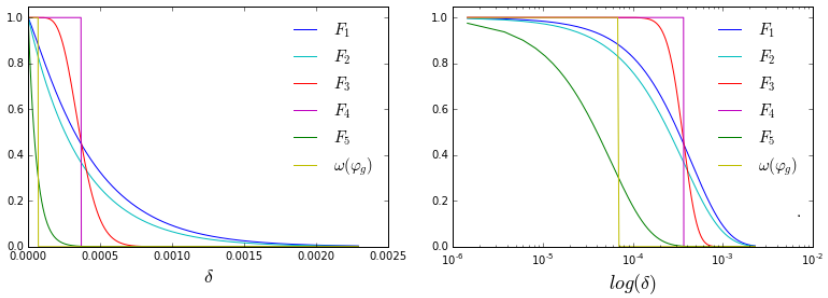


Figure S2: **Various estimations of** $P(\omega(\varphi_g) \geq \delta)$ A single coin is tossed. The probability of tails starts at $\omega(\varphi_0) = 0.5$. Each time a tail is obtained the coin is manipulated to increase the probability of tails that becomes smaller by a dividing factor uniformly sampled between 1 and 4. The number of trials is $j = 30{,}000$. Cumulative distributions $F_1, ..., F_5$ for the different estimations of $\omega(\varphi_g)$ are plot. F_6 (not shown) is equal to F_5. In yellow, the final value of $\omega(\varphi_g)$ that was 0.00007.

we can neglect the multiplicative factors $(h_n\delta + \delta + 1)$ and write:

$$P(\omega(\varphi_g) \geq \delta) \leq (1-\delta)^{\frac{j+1}{g+1}}.$$

For example, for an overestimation of the confidence of, say, at most $K = 10$ and an $\epsilon > 0.005$ our approximation is valid. Figure S1 illustrates the regions where each of the two inequalities hold and show that there is no overlap.

The estimation $F_1 = (1-\delta)^{\frac{j+1}{g+1}} \Pi_{n=0}^{g-1} (h_n\delta + \delta + 1)^{\frac{1}{g+1}}$, its approximation $F_2 = (1-\delta)^{\frac{j+1}{g+1}}$ as well as $F_3 = 1 - I_\delta(g+1, j-g+1)$ transition at the same point $\frac{g+1}{j+2}$ given by the rule of succession of Laplace and are all very similar, with F_2 having the advantage of the simplicity. The estimation based on the regularized incomplete beta function, F_3, transitions very fast and resembles a step function $F_4 = \Theta(\delta - \frac{g+1}{j+2})$. They all overestimate the probability of obtaining tails with the coin φ_g, i.e. the probability of false negative. On the other hand, the estimation based in the final sequence of heads $F_5 = (1-\delta)^{h_g+1}$ tend to better resemble the actual behavior of φ_g provided that h_g is sufficiently large, i.e. when $h_g > \frac{j+1}{g+1}$, which is usually the case. An even better estimation is given by $F_6 = (1-\delta)^{\max(h_g+1, \frac{j+1}{g+1})}$, which behaves as F_2 when $h_g < \frac{j+1}{g+1}$ and as F_5 when $h_g > \frac{j+1}{g+1}$. Figure S3 gives the averaged distributions for the different estimations against the numerically obtained cumulative distribution. In practice we can use:

$$P(\omega(\varphi_g) \geq \delta) = (1-\delta)^{\max(h_g+1, \frac{j+1}{g+1})}$$

and an expected value for the probability of false negative:

$$\overline{\omega}(\varphi_g) = \min\left(\frac{g+1}{j+2}, \frac{1}{h_g+2}\right).$$

C.1.2 Probability of False Negative of Z atoms

The Expected Probability of False Negative. We calculated that the probability of false negative for a single atom is given by the exponential:

$$P(\omega(\phi) \geq \delta) = (1-\delta)^{a(\phi)}.$$

for a positive value $a(\phi) >> 1$:

$$a(\phi) = \max\left(h(\phi) + 1, \frac{j+1}{g(\phi)+1}\right)$$

where $h(\phi)$ is the number of training positive duples used since the moment atom ϕ appears in the model with $g(\phi)$ is the number of links in an inward chain ending at ϕ.

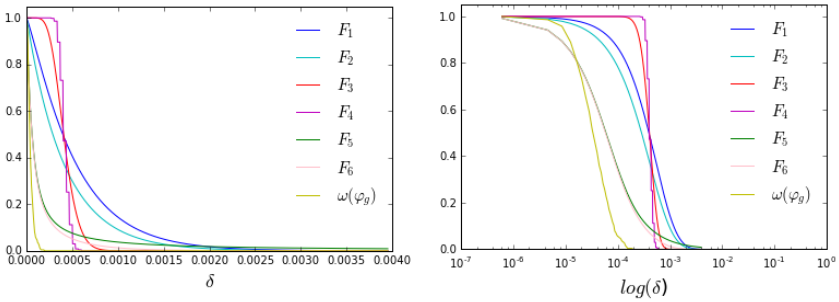


Figure S3: **Average estimations of $P(\omega(\varphi_0) \geq \delta)$ calculated with a simulation.** The probability of tails starts at $\omega(\varphi_0) = 0.5$. Each time a tail is obtained the probability of tails becomes smaller by a dividing factor uniformly sampled between 1 and 4. The number of trials is $j = 30{,}000$ and the experiment is repeated 500 times. Average cumulative distributions for the different estimations $F_1, ..., F_6$ are plotted as well as the actual cumulative distribution of $\omega(\varphi_g)$, in yellow.

The expected value of the probability of false negative, $\omega(\phi_1, ...\phi_Z)$, produced by a model of Z atoms is

$$\overline{\omega}(\phi_1, ...\phi_Z) = \int \omega(\phi_1, ...\phi_Z) \prod_{i=1}^{Z} \frac{dP(\omega_i)}{d\omega_i} d\omega_i$$

where

$$\omega(\phi_1, ...\phi_Z) = 1 - \prod_{i=1}^{Z} (1 - \omega_i)$$

and

$$\frac{dP(\omega_i)}{d\omega_i} = a_i (1 - \omega_i)^{a_i - 1}.$$

The solution of the integral is straightforward:

$$\overline{\omega}(\phi_1, ...\phi_Z) = 1 - \prod_{i=1}^{Z} \frac{a_i}{a_i + 1}.$$

This expression assumes the atoms in the model produce false negatives in an independent manner which overestimates the number of false negatives expected.

At first order, and assuming every a_i is a large number, we can approach the expected probability of false negative with:

$$\overline{\omega}(\phi_1, ...\phi_Z) \approx \sum_{i=1}^{Z} \frac{1}{a_i}.$$

Cumulative distribution function for the PFN of Z atoms. The cumulative distribution for the probability of false negative, $\omega(\phi_1, ...\phi_Z)$, of a model with Z atoms can be computed as the integral:

$$P(\omega(\phi_1, ...\phi_Z) \geq \Delta) = \int_{\omega(\phi_1, ...\phi_Z) \geq \Delta} \prod_{i=1}^{Z} \frac{dP(\omega_i)}{d\omega_i} d\omega_i,$$

where the region of integration goes from $\omega(\phi_1, ...\phi_Z) = \Delta$ to $\omega(\phi_1, ...\phi_Z) = 1$. This integral measures how confident can we be the algebra will indeed behave with a probability of false negative better than Δ.

If we assume all the values of $a(\phi_i)$ are all different, the result of this integral is:

$$P(\omega(\phi_1, ...\phi_Z) \geq \Delta) = \sum_{i=1}^{Z} (1 - \Delta)^{a_i} \prod_{k=1, k \neq i}^{Z} \frac{a_k}{a_k - a_i}.$$

Interestingly, this expression happens to be equal to the value at $x = 0$ of the Lagrange interpolation polynomial passing through the Z points: $(a_1, (1 - \Delta)^{a_1}), ..., (a_Z, (1 - \Delta)^{a_Z})$. The coefficients

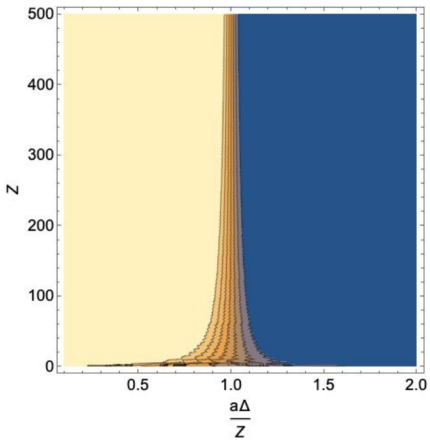


Figure S4: **Probability $P(\omega(\phi_1, ...\phi_Z) \geq \Delta)$ that the probability of false negative produced by Z atoms of equal value of a is greater than a value Δ as a function of $a\tilde{\Delta}/Z$ and Z.** Warmer colors indicate higher probability. Contours are plotted at values of the probability of 0.2, 0.3,0.4,0.5,0.6,0.7 and 0.8. Note the abrupt transition from high to low probability at $\frac{a\tilde{\Delta}}{Z} = 1$ for any value of Z.

$\frac{a_k}{a_k - a_i}$ usually take very large positive and negative values (compared to 1) which makes the expression opaque and difficult to handle. Surprisingly, the coefficients add up to 1:

$$\sum_{i=1}^{Z} \prod_{k=1, k\neq i}^{Z} \frac{a_k}{a_k - a_i} = 1.$$

The result of the integral when the values of $a(\phi_i)$ are all equal for the Z atoms is more illuminating:

$$P(\omega(\phi_1, ...\phi_Z) \geq \Delta) = (1-\Delta)^a \; T_a((1-\Delta)^{-a}, Z-1)$$

where $T_a((1-\Delta)^{-a}, Z-1)$ is the sum of the first $Z-1$ terms of the Taylor series expansion at $a = 0$ of the function $(1-\Delta)^{-a}$ as a function of a. This produces the series:

$$P(\omega(\phi_1, ...\phi_Z) \geq \Delta) = (1-\Delta)^a \sum_{n=0}^{Z-1} \frac{(-a \, \ln(1-\Delta))^n}{n!},$$

that, defining

$$\tilde{\Delta} = -\ln(1-\Delta)$$

becomes

$$P(\omega(\phi_1, ...\phi_Z) \geq \Delta) = \sum_{n=0}^{Z-1} \frac{e^{-a\tilde{\Delta}}(a\tilde{\Delta})^n}{n!},$$

a function of two variables: $a\tilde{\Delta}$ and Z. This expansion corresponds with a regularized gamma function $Q(Z, a\tilde{\Delta})$.

C.1.3 Point of abrupt confidence transition and the expected training set size

Figure S4 shows $P(\omega(\phi_1, ...\phi_Z) \geq \Delta)$ as a function $F(y, Z)$ of the variable $y = \frac{a\tilde{\Delta}}{Z}$:

$$F(y, Z) = \sum_{n=0}^{Z-1} \frac{e^{-yZ}(yZ)^n}{n!}.$$

For $y = 1$ this function takes values close to $1/2$ for all Z, getting even closer to $1/2$ the larger the Z. For a fixed Z the function $F(y, Z)$ transitions from a value very close to 1 to a value very close to 0 abruptly and always at the same point $y = 1$. The transition is more abrupt the larger is Z.

The confidence $P(\omega(\phi_1, ...\phi_Z) \geq \Delta)$ as a function of Δ monotonically decreases, from 1 at $\Delta = 0$ to 0 at $\Delta = 1$. The confidence has an inflection point for any $Z > 1$. At the inflection point,

$P(\omega(\phi_1,....\phi_Z) \geq \Delta) \approx \frac{1}{2}$ so, in practice, we can calculate the inflection point instead of resolving the equation $P(\omega(\phi_1,....\phi_Z) \geq \Delta) = \frac{1}{2}$ which does not lead to a closed formula. The expression for the inflection point i.e. the Δ that makes $\frac{\partial^2 P}{\partial \Delta^2} = 0$, is quite simple:

$$\frac{(a\tilde{\Delta})^Z e^{-a\tilde{\Delta}}(1 + a\tilde{\Delta} - Z)}{Z^2(Z-1)!} = 0.$$

Resolving this equation we get that the inflection point is located at $\tilde{\Delta} = \frac{Z-1}{a}$. At the inflection point, $P(\omega(\phi_1,....\phi_Z) \geq \Delta)$ is equal to $\Gamma(Z, Z-1)/(Z-1)!$, with $\Gamma(z,z)$ the incomplete gamma function, which gives values slightly smaller than $1/2$ and approaches $1/2$ for large Z. For the neighboring value $\tilde{\Delta} = \frac{Z}{a}$ the confidence $P(\omega(\phi_1,....\phi_Z) \geq \tilde{\Delta})$ is equal to $\Gamma(Z,Z)/(Z-1)$ and yields values slightly larger than $1/2$ that also tend to $1/2$ for large Z. Hence, the transition point we are looking for, the Δ that makes $P(\omega(\phi_1,....\phi_Z) \geq \Delta) = \frac{1}{2}$, occurs somewhere between $\frac{Z-1}{a}$ and $\frac{Z}{a}$. In practice, the formula:

$$\frac{Z}{a\tilde{\Delta}} = 1$$

can be used to calculate the transition point and it has the advantage of working also for the case $Z = 1$. Substituting the value of $\frac{1}{a} = \min(\frac{1}{h(\phi)+1}, \frac{g(\phi)+1}{j+1})$ we get the equation for the transition point:

$$\frac{Z}{\tilde{\Delta}} \min\left(\frac{1}{h(\phi)+1}, \frac{g(\phi)+1}{j+1}\right) = 1$$

beyond which $P(\omega(\phi_1,....\phi_Z) \geq \Delta)$ approaches 0 quite rapidly as Δ or j increase. From here we can calculate a bound for the necessary number of training examples:

$$j > \frac{Z\,(g(\phi)+1)}{\tilde{\Delta}},$$

where we have used $h(\phi) < j$, assumed j large and replaced $j+1$ by j.

The abrupt transition of the regularized gamma function implies, in practical terms, that a value Δ (or better) for the probability of false negative can be established with confidence at a particular number of training examples. This confidence changes abruptly from low to high in a narrow window of examples. This does not mean that learning occurs abruptly. Increasingly better probability of false negative values can be gained gradually as the number of training examples increase, but the confidence that we can assign to a given bound Δ for the probability of false negative changes suddenly.

We have derived an expression for the transition point that is valid when all the Z atoms have the same $a(\phi)$. We are interested in the value of Δ at which the transition occurs for sets of atoms with arbitrary values of $a(\phi_i)$, i.e. arbitrary value of $h(\phi_i)$ and $g(\phi_i)$.

For the more general case, with atoms of different $a(\phi_i)$, we can use the formula with the Lagrange polynomials $P(\omega(\phi_1,....\phi_Z) \geq \Delta) = \sum_{i=1}^{Z}(1-\Delta)^{a_i} \prod_{k=1,k\neq i}^{Z} \frac{a_k}{a_k - a_i}$. We show at the end of this section that the transition occurs for this cumulative distribution at:

$$\frac{1}{\tilde{\Delta}} \sum_{i=1}^{Z} \frac{1}{a_i} = 1,$$

which generalizes our previous result $\frac{Z}{a\tilde{\Delta}} = 1$ for Z atoms with the same exponent a. Substituting the value of each a_i we get the equation for the transition point:

$$\frac{L}{\tilde{\Delta}} = 1 \quad \text{where} \quad L = \sum_{i=1}^{Z} \min\left(\frac{1}{h(\phi_i)+1}, \frac{g(\phi_i)+1}{j+1}\right).$$

Taking into account $\tilde{\Delta} = -\ln(1-\Delta)$, if we solve for Δ:

$$\Delta_T = 1 - e^{-\sum_{i=1}^{Z} \frac{1}{a_i}}.$$

we obtain the value of the transition point Δ_T. This formula and the formula above for the average $\overline{\omega}(\phi_1, ...\phi_Z) = 1 - \prod_{i=1}^{Z} \frac{a_i}{a_i+1}$ are numerically very close (just do a first order approximation). We have:

$$\Delta_T \approx \overline{\omega}(\phi_1, ...\phi_Z),$$

which occurs because most of the weight for the integral yielding the average distribution is located at the narrow transition region. Substituting the value of a_i

$$\overline{\omega}(\phi_1, ...\phi_Z) = 1 - e^{-L}.$$

We refer to L as "the load", a value that can be computed in practice and can be used to determine, much before it can be established using a test set, how close the algebra is to learning the problem at hand. The load measures the immaturity of the atoms and how much extra learning is still required. The larger is the load, the more far away the model is from producing a good probability of false negative. The load can take values much larger than 1 at the beginning of learning and, as the algebra learns, it goes down. The algebra produces small values for the probability of false negative when the load is smaller than 1. For a pattern recognition problem the load tends to first increase and then decrease until it reaches the region with values $L \leq 1$. It is only in the region $L \leq 1$ that the amount of learning can be measured with a test set. When the load is larger than 1 the algebra produces a 100% false negatives in the test set. In this region learning occurs at constant probability of false negative equal to 1, but can be measured using L.

Solving for j we get the number of training examples needed to get a probability of false negative better than Δ or, more precisely, to get the number of training examples needed to go beyond the transition point for Δ:

$$j \geq \frac{\sum_{i=1}^{Z}(g(\phi_i)+1)}{\tilde{\Delta}}.$$

where, again, we have assumed j large and replaced $j+1$ by j.

C.1.4 General case for the transition point

We are going to do a calculation of the transition point of $P(\omega(\phi_1, ...\phi_Z) \geq \Delta)$ for Z atoms ϕ_i with different exponents a_i. First notice that, still in this more general case, $P(\omega(\phi_1, ...\phi_Z) \geq \Delta)$ is a function of $\tilde{\Delta}\mathbf{a}$ rather than a function of Δ and $\mathbf{a}$, where we have used $\mathbf{a}$, in bold, to represent the vector with Z components, a_i. We can indeed rewrite the confidence as:

$$P(\omega(\phi_1, ...\phi_Z) \geq \Delta) = \sum_{i=1}^{Z} e^{\tilde{\Delta}a_i} \prod_{k=1,k\neq i}^{Z} \frac{\tilde{\Delta}a_k}{\tilde{\Delta}a_k - \tilde{\Delta}a_i}.$$

Its derivative is:

$$\frac{dP(\omega(\phi_1, ...\phi_Z) \geq \Delta)}{d\Delta} = \frac{d\tilde{\Delta}}{d\Delta}\left(\prod_{j=1}^{Z} a_j\right)\sum_{i=1}^{Z}\left(e^{\tilde{\Delta}a_i}\prod_{k=1,k\neq i}^{Z}\frac{1}{a_k - a_i}\right),$$

which is not a function of $\tilde{\Delta}\mathbf{a}$, but the closely related function:

$$G(\tilde{\Delta}\mathbf{a}) \equiv -\left(\prod_{j=1}^{Z} \tilde{\Delta}a_j\right)\sum_{i=1}^{Z}\left(e^{-\tilde{\Delta}a_i}\prod_{k=1,k\neq i}^{Z}\frac{1}{\tilde{\Delta}a_k - \tilde{\Delta}a_i}\right)$$

is, and it can be used to calculate the derivative as:

$$\frac{dP(\omega(\phi_1, ...\phi_Z) \geq \Delta)}{d\Delta} = G(\tilde{\Delta}\mathbf{a})\frac{1}{\tilde{\Delta}}\frac{d\tilde{\Delta}}{d\Delta}.$$

The equation $P(\omega(\phi_1, ...\phi_Z) \geq \Delta) = 0.5$ yields the location at which the confidence transitions but it cannot be resolved. We could try, as we did before, to search for the inflection point but this time it does not lead to a closed expression. Fortunately, it is enough with finding a value in the narrow region where the cumulative distribution transitions. Any value on this region works for us.

We are going to use a convenient translational property of $G(\tilde{\Delta}\mathbf{a})$. If we add the same value x to each of the components of the vector $\tilde{\Delta}\mathbf{a}$ we get:

$$G(\tilde{\Delta}\mathbf{a}+x)=\frac{\prod_{j=1}^{Z}(\tilde{\Delta}a_j+x)}{\prod_{j=1}^{Z}\tilde{\Delta}a_j}e^{-x}G(\tilde{\Delta}\mathbf{a}).$$

Let us shift the vector $\tilde{\Delta}\mathbf{a}$ by adding x to each of its components in order to find a value of x that minimizes $\frac{dP(\omega(\phi_1,....\phi_Z)\geq\Delta)}{d\Delta}$. The value of the transition can be found by locating the stationary point of $G(\tilde{\Delta}\mathbf{a})$:

$$\frac{dG(\tilde{\Delta}\mathbf{a}+x)}{dx}=0$$

from which we get:

$$\left(-\prod_{j=1}^{Z}(\tilde{\Delta}a_j+x)+\sum_{i=1}^{Z}\prod_{j=1,i\neq j}^{Z}(\tilde{\Delta}a_j+x)\right)\frac{e^{-x}G(\tilde{\Delta}\mathbf{a})}{\prod_{j=1}^{Z}\tilde{\Delta}a_j}=0,$$

and, finally: $\sum_{i=1}^{Z}\frac{1}{\tilde{\Delta}a_i+x}=1$, which locates the transition of $P(\omega(\phi_1,....\phi_Z)\geq\Delta)$ at

$$\frac{1}{\tilde{\Delta}}\sum_{i=1}^{Z}\frac{1}{a_i}=1.$$

C.2 Generalizing Subsets selected by Sparse Crossing

The freest model, $F_C(R^+)$, of a set R^+ of atomic sentences is usually too large to compute. However, using Sparse Crossing, it is possible to directly calculate a subset of its atoms without the need of computing the entire freest model first. For that to be possible, a set of negative duples, R^-, is also required. In a classification problem, counterexamples can be provided as negative duples.

The subset of atoms computed by sparse-crossing at batch i is such that it satisfies $R_i=R_i^+\cup R_i^-$ with just the necessary atoms needed to discriminate R_i^-, i.e., at most $|R_i^-|$ atoms. This results in a small subset of atoms that are good at discriminating the negative duples. Moreover, the less correlation the atoms in the subset have, the fewer atoms are needed to discriminate R_i^-, so Sparse Crossing also tends to produce atoms that are relatively uncorrelated.

In addition to the "master" model of small cardinality obtained from the sparse-crossing of a batch, a "union model" obtained as the union of the master models of all previous batches is also computed (see Appendix section D). The union model undergoes a sieve in which atoms inconsistent with $\cup_{k=0}^{i}R_k^+$ are removed resulting in a model that satisfies this set of positive duples. The union model is atomized by a subset of atoms of the freest model of $\cup_{k=0}^{i}R_k^+$, and is typically much larger than the master model but still very small compared to the size of the freest model.

The mental experiment of Section C can be done in practice using Sparse Crossing. In this experiment, we study how well subsets of the model $F_j=F_C(s_1^+,s_2^+,...,s_j^+)$ approximate a hypothetical model M of perfect performance. Assume the first i batches are such that $\cup_{k=0}^{i}R_k^+=\{s_1^+,s_2^+,...,s_j^+\}$. To study the evolution of the expected PFP and PFN we can compute subsets of atoms of F_j of various sizes by selecting atoms from the union model. We can extract from the union model subsets of any size of discriminative, low-correlation atoms, for which we use the following selection method: we select a small subset of atoms that suffices to discriminate the training set or part of it and, by adding multiple such subsets of smallest cardinality we can extract larger subsets of F_j of any size up to the size of the union model.

Figures S5 and S6 show the observed $\overline{\text{PFP}}$ and $\overline{\text{PFN}}$ for the problem of identifying vertical bars in a noisy background and for the problem of handwritten digits, respectively. For the first problem there is a very large supply of examples while for the second, the number of examples is limited to 50K. The calculations of the expected PFP and PFN assumes a sequence $s_1^+,s_2^+,s_3^+,...,s_j^+$ of different positive examples, which is what happens in the first problem. However, for the second problem the sequence of examples contains repeated samples and after a few batches, only contains repeated examples. While with Full Crossing training using repeated examples has no effect, with Sparse

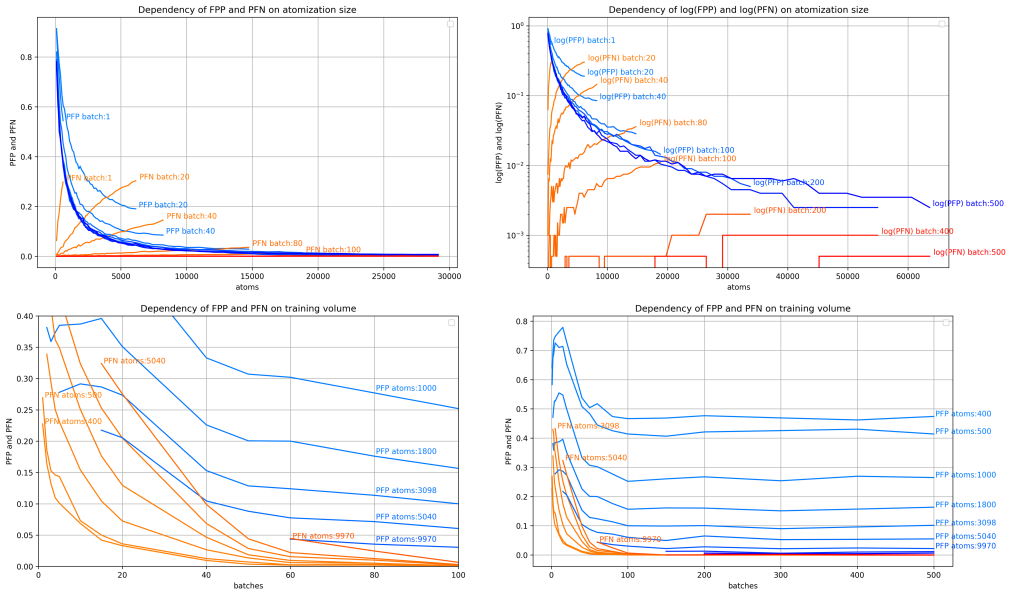


Figure S5: **Evolution of the** PFP **and** PFN **for the problem of classifying images with or without a complete vertical black bar in an** 8×8 **grid in the presence of** 50% **background noise**. Every batch consisted of $1{,}000$ examples and $1{,}000$ counterexample images. As expected, the PFN (in orange/red) increases linearly with the number of atoms in the model, with a slight sublinear growth due to atom correlations. For the PFP (in blue), we observe a very fast geometric decrease with the number of atoms, with diminishing rate due to correlations. As training progresses, an strong decrease of both the PFP and PFN is observed for models with the same number of atoms. After about 100 batches, most atoms reach maturity; after that, the larger the subset of atoms the better the performance. However, more training does not result in better performance for sets of atoms with the same size. The models were obtained using Sparse Crossing, which computes a model M_b at batch b, that is a subset of the freest model of the data. At different stages of the training, subsets of M_b with good discriminative capacity and low correlation were obtained to calculate the figures. Discriminative, low-correlation subsets were calculated by selecting a set of atoms, as small as possible, that collectively sufficed to discriminate $5{,}000$ counterexamples. To obtain larger subsets multiple subset selections were carried out, with replacement, and using the same set of counterexamples. The resulting subsets were added. The performance of the subsets was measured using a test set of $2{,}000$ examples and $2{,}000$ counterexamples.

Crossing it is useful as it leads to the discovery of more and more discriminative atoms of the freest model as explained in Section D.8 or Appendix D.

The experimental evolution of the $\overline{\text{PFP}}$ and $\overline{\text{PFN}}$ in both problems matches very well the theoretical calculations. However, for MNIST, it does so only up to the batch 50 or so, after which most examples are repeated and our theoretical assumptions no longer hold.

The experimental result using repeated examples in MNIST is shown in Figure S6. A model for MNIST was calculated training for 900 batches using Sparse Crossing. The number of examples in a training batch grew linearly from 500 examples in batch 1 to two/thirds of the training set at batch 500. After batch 500, batch size remained constant until batch 900. In Figure S6A, we can see that the experimental PFP behaves similar to that expected for training with non-repeated data, with a very strong decrease with the number of atoms and the amount of training. Since the expected PFN decreases geometrically with Z, as Sparse Crossing discovers more and more atoms of F_j it is not surprising that we see a strong decrease with the size of the atom subset despite training occurring with the same data. For the experimental PFP, we observe, as expected, an increase with the number of atoms that flattens quite rapidly. However, here we observe that the PFN worsens with the amount of training. Narrower atoms discovered by Sparse Crossing are more effective at correctly discriminating negative duples but they also have a higher risk of causing a false negative. In addition,

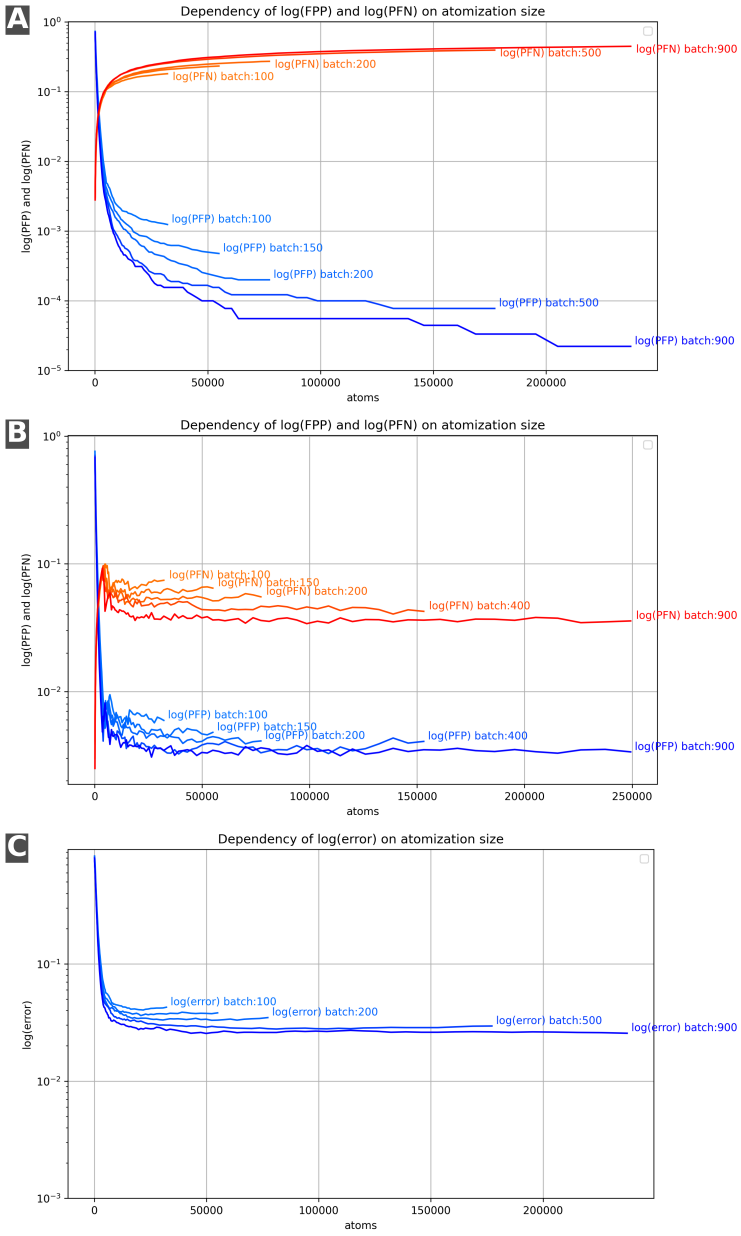


Figure S6: **Repeated training with the same data for the MNIST handwritten digit classification.** The evolution of the PFP and PFN is shown in plot A. In plot B, the PFP and PFN calculated from an optimal number of atom misses (beyond which a duple is considered negative) estimated using the validation dataset. Plot C, corresponds to the classification error rate obtained by selecting the class with fewer misses, without using the validation. The figure illustrates that Sparse Crossing produces models that improve with the amount of training even if the same data is presented 490 times, whiteout overfitting, and given constant results across a wide range of atom subset sizes.

the theoretical $\overline{\text{PFP}}$ only decreases with the number of different positive duples presented to the algebra, so we expect the PFP to be affected by the presence of repeated examples.

In this situation, after around batch 100, we have a PFN that improves with training and a PFP that worsens, so we should wonder if the algebra is still learning or not. To answer this question

consider Figure S6B. Now we allow a few misses before a duple is considered negative instead of just 1 miss, in order to evaluate a subset we used a validation dataset of the same size to determine the optimal number of atom misses needed to declare a duple negative. We can see now two important differences: first, both the PFP and PFN improve with the amount of training and, second, results no longer depend upon subset size for subsets of more than 30K atoms. In Figure S6C we use a simpler heuristic; to classify a sample we selected the class with fewer misses. This does not require a validation dataset and, since there is a false negative for each false positive, PFP and PFN both become equal to the classification error rate, which reached about 2.5%. Again we observe the same as before; error decreases with training and is independent of subset size, actually flat, in a wide range of sizes beyond 30K atoms. After about 100 epochs the error in the training set 0, but this does not stop learning, as it continues for hundreds of epochs and, probably, for much longer. Learning continues without overfitting even after the training set have been presented 490 times. There is a clear performance improvement for subsets of the same size as training progresses using the same data. This is indication that Sparse Crossing is able to find increasingly better atoms with more training. The mechanism involves leveraging previously discovered atoms to find narrower atoms (with smaller upper constant segments) and non-redundant atoms of the model F_j.

D Sparse crossing

The Full Crossing operator can be used to construct the freest semilattice that satisfies a given set of axioms. These axioms, denoted as $R = R^+ \cup R^-$, consist of atomic sentences (R^+) and negated atomic sentences (R^-) without quantifiers. The computation of the freest model of R necessitates a full-crossing operation for each atomic sentence in R^+.

The freest semilattice model satisfying a set of axioms typically has a substantial number of subdirectly irreducible components. Consequently, it is often impractical to first calculate the freest model and then identify a generalizing subset, as the freest model is computationally intractable. To address this challenge, we employ a sparse version of the Full Crossing operator. Sparse Crossing retains the atoms required to maintain the satisfaction of negative axioms while selectively eliminating other atoms.

The atom elimination process is guided by the following considerations: to discover generalizing models, we aim to identify atoms of the freest model of the positive axioms that most effectively discriminate the negative axioms (see Definition 13). We also must avoid discarding atoms that, although removable at a particular crossing step without affecting the negative axioms, are essential for maintaining the satisfaction of negative axioms in subsequent crossing steps.

D.1 Building free models iteratively

Besides sparsity, several other differences exist between Full and Sparse Crossing. First, while Sparse Crossing utilizes R^-, Full Crossing does not, as every negative atomic sentence consistent with R^+ is always satisfied by the freest model of R^+. Second, Sparse Crossing typically employs batches, breaking R into subsets (often with replacement) and proceeding iteratively, batch by batch, in a process resembling neural network training. In contrast, Full Crossing performs a crossing operation for each axiom of R^+ in any order, obtaining the final model after crossing each positive tuple exactly once. An iterative method is necessary because discovering the most discriminative (useful) atoms of the freest model of R^+ in a single attempt is unlikely to occur.

The processing of a batch i with axioms $R_i = R_i^+ \cup R_i^-$ starts with a model M_{i-1} and ends with a model M_i that satisfies R_i. We thus obtain a sequence of atomized models $M_0, M_1, \ldots$. The sequence starts with an initial model M_0, usually equal to the freest semilattice over the empty set of axioms, $M_0 = F_C(\emptyset)$. The model $F_C(\emptyset)$ has one atom for each constant in C, so it can be easily computed (see Appendix A.2, Theorem 13). We refer to the models M_i as "master models".

After each master model M_i is calculated, we add it to a model that contains the result of processing previous batches, generating a second sequence of models $\{N_0, N_1, \ldots\}$. We refer to these models as "union models" The model $N_{i-1} + M_i$ always satisfies R_i^-; however, it does not necessarily satisfy R_i^+. To resolve this, we discard the atoms of N_{i-1} that are not consistent with R_i^+:

$$N_i = \{\phi \in N_{i-1} : \phi \models R_i^+\} \cup M_i.$$

Since $\{\phi \in N_{i-1} : \phi \models R_i^+\} \models R_i^+$ and $M_i \models R_i^+$, then $N_i \models R_i^+$. Since $M_i \models R_i^-$ and the atoms of M_i are atoms of N_i then N_i satisfies R_i^-. It follows, $N_i \models R_i$.

As batches are processed, the union models become more consistent with R, usually larger, and always composed of atoms that are more effective at discriminating the axioms of R^-. After a number of batches, the union models satisfy the entire set of axioms R (the training set), and the training reaches a point after which no atoms are removed and union models only grow larger. We can say that, at that point, all the atoms in the union model have reached full maturity.

D.2 Keeping model size under control

The full-crossing (see Definition 15) of a duple $r = (r_L, r_R) \in R^+$ over a model M involves removing the atoms of the discriminant $dis_M(r)$ and then adding the atoms in $dis_M(r) \bigtriangledown L_M^a(r_R)$ where the lower atomic segment $L_M^a(r_R) = \{\varphi \in M : \varphi < r_R\}$ is the set with the atoms of the right-hand side r_R of the duple. The full-crossing operation transforms a model M into the model:

$$\Box_r M = (M - dis_M(r)) + dis_M(r) \bigtriangledown L_M^a(r_R)$$

where we have used $-$ for set subtraction and $+$ for set union.

We can view a full-crossing operation as an atom replacement process. An atom $\phi \in dis_M(r)$ is replaced by multiple atoms $\phi \bigtriangledown \psi_k$, where ψ_k are the atoms in the lower atomic segment $L^a(r_R)$. Specifically, an atom ϕ in the discriminant is replaced by the set of atoms:

$$\phi \to \{\phi \bigtriangledown \psi_k : \psi_k \in L^a(r_R)\}.$$

To ensure that atom discarding allows for the satisfaction of all negative axioms, even after subsequent crossing steps, Sparse Crossing relies on an invariant called the trace. The trace is a function that maps an element of an atomized semilattice model (either an atom or a regular element) to a set. The rationale behind this approach is that we can eliminate an atom generated by Full Crossing as long as we maintain the trace of all elements in the model unchanged.

The trace function (see Theorem 27) has the property:

$$Tr(\phi) = \cap_{\psi_k \in L^a_M(r_R)} Tr(\phi \bigtriangledown \psi_k),$$

so it is possible to discard some of the atoms $\phi \bigtriangledown \psi_k$ in the right-hand side if the trace $Tr(\phi)$ given by this expression remains the same. In other words, instead of the replacement above, Sparse Crossing uses:

$$\phi \to \{\phi \bigtriangledown \psi_k : \psi_k \in S_\phi \subseteq L^a_M(r_R)\},$$

where S_ϕ is any subset of $L^a_M(r_R)$ that satisfies

$$Tr(\phi) = \cap_{\psi_k \in S_\phi} Tr(\phi \bigtriangledown \psi_k).$$

This replacement operation keeps the traces of every element of the atomized semilattice unaltered as proven in Theorem 28, and corresponds with Algorithm 7 in the pseudocode.

Assume a model $M \models R_i^-$. The sparse-crossing of a duple r over M transforms M in a model M' that satisfies r^+ and R_i^- and it is atomized by a subset of the atoms of $\square_r M$, i.e. it is less free than $\square_r M$ since the atoms of M' are all atoms of $\square_r M$.

D.3 Building the dual

Before calculating the sparse-crossing of the positive tuples in R_i^+ of batch i, we must first determine an appropriate codomain for the trace function. We refer to such codomain as "the dual".

Consider the set of constants and terms:

$$E_i = C \cup Terms(R_i^+ \cup R_i^-) \cup \{T_\phi : \phi \in N_{i-1} \wedge \phi \models R_i^+\}$$

where T_ϕ is the pinning term of an atom ϕ, i.e., the term with component constants $C - U^c(\phi)$, and:

$$Terms(R_i) = \{r_L : r \in R_i\} \cup \{r_R : r \in R_i\}$$

are the terms mentioned in the set of axioms R_i of the batch i.

Let the dual D_i be an atomized semilattice generated by a set of constants C^* with cardinality $|C^*| = |E_i|$. The set C^* contains a constant for each constant of C, a constant for each term mentioned in $R_i^+ \cup R_i^-$ and a constant for each pinning term T_ϕ of the union model N_{i-1} after it has been filtered out using R_i^+. We denote the constant in C^* corresponding to the element $x \in E_i$ with the symbol $[x]$, which we refer to as "the dual of x."

The dual D_i satisfies the dual of the axioms of R_i:

$$(r_L, r_R) \in R_i \to ([r_R], [r_L]) \in R_i^*,$$

i.e. the axioms $R^* = R_i^{*+} \cup R_i^{*-}$.

Since every element mentioned in R_i^* is a constant of D_i, atomizing the dual is very easy; it is enough with adding an atom ξ_c to each constant c of C^*, building a directed graph connecting the constants and then calculating the transitive closure:

The formal definition of the dual is given in Definition 19, and the method to compute it is presented in Section D.9.

An important result is that the dual D_i satisfies R_i^{*-} if and only if R_i is a consistent set of axioms (see Theorem 30). The fact that D_i can be atomized if and only if the axioms R_i are consistent is particularly advantageous as it provides a consistency check at no additional computational cost.

Optionally, we can discard atoms in the dual as long as R_i^{*-} remains satisfied, which can always be achieved with no more than $|R_i^-|$ atoms (see Theorem 39).

Theorem 30 demonstrates that D_i is the dual semilattice of the intersection of $F_C(R_i^+)$ with E_i. Furthermore, D_i is equal to $F_{E_i}(R_i^{*+})$. It is important to note that although we refer to D_i as "the dual", it only corresponds to a small subset of the dual of $F_C(R_i^+)$.

D.4 Building the trace

Once the dual D_i is constructed it is possible to build a trace function for batch i. Given a dual D, the trace is a function that maps elements of a model M to a subset of atoms of D. To calculate the trace of any atom over C, we must use:

$$Tr(\phi) = \{\xi : \exists c \in C,\ (\phi < c) \wedge (D \models \xi < [c])\}$$

where $c \in C$ is a constant and ξ is an atom of the dual. It follows that $Tr(\phi) = \cup_{c \in U^c(\phi)} L^a([c])$, where we write $L^a([c])$ instead of $L_D^a([c])$ for brevity, as there is no ambiguity. This formula corresponds to Algorithm 2. The trace of a union of atoms satisfies (see Theorem 25):

$$Tr(\phi \bigtriangledown \psi) = Tr(\phi) \cup Tr(\psi).$$

The trace of an atom is universal, i.e., it is the same for any model and depends solely on the dual. Given the traces of the atoms, we can calculate the trace of a regular element (i.e., a constant or an idempotent summation of constants) in a model M using:

$$Tr_M(x) = \cap_{\phi \in L_M^a(x)} Tr(\phi),$$

which is what Algorithm 3 does. Unlike the traces of atoms, the traces of regular elements depend on the specific model M and are therefore not universal. However, in Theorem 33, we demonstrate that the traces of regular elements are well-defined, meaning they are determined by the model M itself and not by a particular atomization of M. In general, and as long as there is no ambiguity, we omit the model subindex from the notation and write $Tr(x)$ for the trace of a regular element.

D.5 Trace constraints and Λ_i

Before initiating the sparse-crossing of the duples in R_i^+, we must ensure that our initial model satisfies R_i^-. This can always be achieved by adding to the model M_{i-1}, obtained from the previous batch, a set of atoms Λ_i; in this section we will learn how to compute Λ_i.

We begin by calculating the trace of every atom in our initial model M_{i-1}. Using these atom traces, we compute the traces of the terms in R_i and ensure that the following two sets of constraints are satisfied:

$$r \in R_i^+ \rightarrow\ Tr(r_R) \subseteq Tr(r_L),$$
$$r \in R_i^- \rightarrow\ Tr(r_R) \not\subseteq Tr(r_L).$$

We refer to these sets of constraints as positive and negative trace constraints, respectively.

The negative trace constraints are necessary to ensure that R_i^- holds at all times. According to Theorem 29, as long as a model satisfies $T(b) \not\subseteq T(a)$, then $M \models a \not\leq b$. It follows that one or multiple trace-preserving operations on a model that satisfies $T(b) \not\subseteq T(a)$ will always yield a model that satisfies $a \not\leq b$. Since all the operations we use are trace-invariant, we have guaranteed that at the end of the crossing phase, all positive and negative axioms are satisfied.

Positive trace constraints are necessary as they imply that an atom in the discriminant has a trace $Tr(\phi)$ equal to the intersection of the traces $Tr(\phi \bigtriangledown \psi_k)$ of the atoms generated during the crossing (see Theorem 27). If the positive trace constraints are not satisfied, the crossing operation is not trace-invariant.

Theorem 35 demonstrates that both sets of trace constraints can be satisfied by merely adding new atoms to M_{i-1}, each with a single constant in its upper constant segment. In fact, adding one atom for each constant of C always produces a model that satisfies the trace constraints (see Theorems 34 and 33). However, we aim to minimize the number of added atoms, as an increase in atoms leads to more computational work. Trace constraints can typically be satisfied by adding only a few atoms to

M_{i-1}. As shown in the proof of Theorem 35, a negative trace constraint for a duple $r \in R^-$ that is not satisfied can always be rectified by adding new atoms, each under a single constant in the lower segment of r_L. Similarly, a positive trace constraint for a duple $r \in R^+$ can be enforced by adding new atoms, each under a single constant, in the lower segment of r_R.

The procedure to compute the new atoms that ensure all trace constraints are satisfied, the set Λ_i, is presented in pseudocode in Algorithm 5. To understand why this works see the proof of Theorem 35.

D.6 A note on duality and trace constraints

To better understand the role played by the trace constraints consider the following. Restricted to the set E_i, there is a duality given by $\cup \leftrightarrow \cap$ and $L^a_{F_C(R_i^+)}(t) \leftrightarrow L^a_{D_i}([t])$. To be more specific, for $a, b \in E_i$ we have $F_C(R_i^+) \models a \leq b$ if and only if $L^a([b]) \subseteq L^a([a])$ where we have omitted the subindex in $L^a_{D_i}$ for less clutter. We also have $L^a([b]) = \cap_{c \in C(b)} L^a([c])$ (see Theorem 31). We saw in Section D.3 that D_i corresponds, via duality, to a small subset E_i of $F_C(R_i^+)$.

This duality extends beyond the small subset E_i through the trace. For any two terms of any semilattice M, we have that $M \models a \leq b$ implies $Tr(b) \subseteq Tr(a)$, which holds true even if the trace constraints are not satisfied. This duality is weaker for M than for the dual D_i, as it only operates from left to right; however, it is also stronger as it is not restricted to E_i and applies to all elements of M. If the positive trace constraints hold for M, we also have $F_C(R_i^+) \models a \leq b$ implies $Tr(b) \subseteq Tr(a)$. Furthermore, when the positive trace constraints hold for M, for any two terms a, b such that $\Box_{R^+} M \models a \leq b$, the trace in M satisfies $Tr(b) \subseteq Tr(a)$ (although the implication from right to left is not necessarily true). This is demonstrated in Theorem 36.

D.7 Trace-invariant simplification of the master

The trace is linear with respect to the idempotent operator (see Theorem 25):

$$Tr(a \odot b) = Tr(a) \cap Tr(b).$$

This property also implies that a regular element x of the atomized semilattice:

$$Tr(x) = \cap_{c \in C(x)} Tr(c).$$

where $C(x)$ is the set of component constants of x (see also Theorem 32).

By construction, sparse-crossing a duple is a trace-invariant operation as it leaves the traces of all elements unchanged. Since the trace of a term is equal to the intersection of the traces of its component constants we can build another trace-invariant operation; removing atoms while leaving the trace of every constant unaltered shall be trace-invariant.

Discarding atoms while preserving the traces of the constants is a relatively efficient and effective operation that can be applied at any time. This operation requires iterating through the constants of the model rather than through the elements mentioned in the axioms. This simplification procedure is implemented in Algorithm 8.

A master model that satisfies the trace constraints can be computed with a size that depends only on the size of the dual (see Theorem 40). Since the atomization of the dual requires no more than $|R^-|$ atoms, we have a guarantee that, by using trace-invariant simplification and sparse-crossing, the master model maintains a manageable size at all times.

D.8 Crossing smarter every batch

The pinning terms:

$$T_\phi : \phi \in N_{i-1} \wedge \phi \models R_i^+$$

are present in the set E_i, which determines the constants of the dual. However, a dual could be constructed without pinning terms, using the smaller set $E_i = C \cup Terms(R_i^+ \cup R_i^-)$, and a valid trace would be obtained, allowing for the calculation of Sparse Crossing.

By including the pinning terms in E_i, we allow the atoms learned in previous batches to influence the crossing process. The effect of adding the pining terms of (the atoms of) the union model N_{i-1} is faster convergence and better master models.

The atomization of the dual can be simplified by discarding atoms, following the straightforward procedure discussed in the proof of Theorem 39. This simplification becomes particularly effective when pinning terms are taken into account, often resulting in a dual with significantly fewer than $|R_i^-|$ atoms. The size of the master can be bounded by $\alpha|D|$, where α is the average size of an atom in N_{i-1} and $|D|$ is the number of atoms in the dual. A smaller dual not only leads to a smaller master M_i (see Theorem 40) but also accelerates its calculation.

Theorem 38 (ii) clarifies the guiding role played by the pinning terms. There is a pinning term T_ψ for each atom $\psi \in N_{i-1}$. This Theorem states that, given any atom ϕ over C, the atom ξ_ψ of the dual that is edged to $[T_\psi]$ satisfies:

$$\xi_\psi \notin Tr(\phi) \text{ if and only if } U^c(\phi) \subseteq U^c(\psi).$$

Assume ϕ is replaced with $\phi \rightarrow \{\phi \bigtriangledown \varphi_k : \varphi_k \in S_\phi \subseteq L_M^a(r_R)\}$ where the set S_ϕ is selected so the replacement is trace-invariant. Since $Tr(\phi \bigtriangledown \varphi_k) = Tr(\phi) \cup Tr(\varphi_k)$:

$$\xi_\psi \notin Tr(\phi) \cup Tr(\varphi_k) \text{ if and only if } U^c(\phi \bigtriangledown \varphi_k) \subseteq U^c(\psi).$$

The invariance of the trace, $Tr(\phi) = \cap_{\varphi_k \in S_\phi}, Tr(\phi \bigtriangledown \varphi_k)$, requires that for each $\xi_\psi \notin Tr(\phi)$, there exists at least one $\varphi_k \in S_\phi$ such that $\xi_\psi \notin Tr(\varphi_k)$. It follows that when $U^c(\phi) \subseteq U^c(\psi)$, the atom ϕ is replaced by at least one atom with $U^c(\phi \bigtriangledown \varphi_k) \subseteq U^c(\psi)$. This must occur for the potentially numerous atoms ψ that satisfy $\xi_\psi \notin Tr(\phi)$. Since the selection of the atoms $\varphi_k \in S_\phi$ is simultaneously trace-invariant and of minimal cardinality (minimal in a best-effort sense), the selection results in atoms $\phi \bigtriangledown \varphi_k$ with upper constant segments that are, often, subsets of $U^c(\psi)$ for many atoms $\psi \in N_{i-1}$.

This mechanism generates atoms that are common components of previously learned atoms, which explains why non-redundant atoms can be discovered from the pinning terms of redundant atoms. The inclusion of pinning terms results in models that are not only consistent with R_i but also more consistent with previous batches and, consequently, more consistent with R, an observation supported by experimental evidence. Without pinning terms in the dual, the union of master models still approximates the freest model of the axioms as training progresses; however, learning may be significantly slower.

D.9 Efficient computation of the dual

The construction of the dual can be carried out as described in Definition 19, which involves constructing a graph and computing its transitive closure. Although this formal definition is useful and facilitates Theorem proving, graph manipulations can be computationally expensive and are not necessary.

We present here a simple method that involves a few stages, all parallelizable. This method corresponds to Algorithm 4, and it is based on computing "indicator sets" that become atoms of the dual at the end of the computation:

1 - For each term at the right hand side of any duple in R^- create an "indicator set" equal to component constants set of the term.

2 - For each duple $r = (r_L, r_R)$ in R^+ and for each indicator set, if the set of component constants of r_R, the set $C(r_R)$, is a subset of the indicator set, add $C(r_L)$ to the indicator set.

3 - Repeat step 3 until not indicator set undergoes further changes.

4 - Create a vector v_1 with the indicator sets, in any order. Make sure each indicator set only appears once by removing repeated indicator sets.

5 - Transform the set of atoms $\phi \in N_{i-1}$ into a vector v_2, in any order. Concatenate vectors v_1 and v_2 in a single vector v. Each position (an index) of the vector v becomes the index of an atom of the dual.

6 - For each element s in the set $C \cup Terms(R_i^+ \cup R_i^-)$ compute the lower atomic segment of its dual as follows: the lower atomic segment of $[s]$ contains the atom of the dual corresponding to position i of vector v_1 if the set of component constants of s is a subset of the indicator set $v_1(i)$. In addition, the lower atomic segment $[s]$ contains the atom with index $|v_1| + i$ if the set of component constants of s is disjoint with $U^c(v_2(i))$.

7 - Optionally, reduce the set of atoms of the dual as in Theorem 39.

The upper constant segments of the atoms of the dual are not explicitly computed because they are never needed; it is enough with knowing the lower atomic segment of the elements in the set $C \cup Terms(R_i^+ \cup R_i^-)$.

The method assumes that every pinning term T_ψ corresponds to an atom ψ consistent with R_i^+. Theorem 37 states that for such terms, every constant c for which $[c]$ is in the upper segment of $[T_\psi]$ is a component constant of T_ψ. Consequently, there is no need to apply Step 2 to the pinning terms as it would have no effect on the trace function.

A potential alternative approach to building a dual is given by Haidar et al. in [45].

D.10 Putting all together

The main loop of Sparse Crossing corresponds to Algorithm 1. The input consists of a set of axioms R and an initial model M_0. If no M_0 is provided, the freest model $M_0 = F_C(\emptyset)$ can be used. The axioms may be divided into any number of batches R_i, with or without replacement, typically with replacement.

Initialize the union model $N_0 = M_0$. For each batch:

Step 1: Trim the union model N_{i-1} removing the atoms inconsistent with R_i^+.

Step 2: Build the dual D_i.

Step 3: Use D_i to calculate the trace of every atom in M_{i-1}. Using these traces, compute and evaluate each trace constraint. For each unsatisfied constraint, add atoms to M_{i-1} until the constraint is satisfied. Repeat this step as necessary (typically twice) until all trace constraints are satisfied.

Step 4: Calculate the sparse-crossing of every duple in R_i^+. The resulting model after enforcing all positive duples is M_i, which satisfies R_i. If at any point the model exceeds a predetermined threshold, discard atoms while maintaining the traces of all constants invariant.

Step 5: Obtain N_i as the union of M_i with the trimmed union model N_{i-1}.

D.11 Definitions

Definition 19. Given the set E_i of terms and the axioms of the dual R_i^{*+} as in Section D.3, let G be the graph with the following edges:

1 - a directed edge, $[r_R] \to [r_L]$, for each duple in R_i^{*+}

2 - directed edges $[x] \to [c]$ for each component constant c of every term x in E_i.

3 - a directed edge $[y] \to [x]$ between any two elements x and y of E_i such that $C(x) \subseteq EC(y)$. Here, $C(x)$ is the set of component constants of x and $EC(y)$ is a superset of $C(y)$, the "extended set" $EC(y) = \{y\} \cup \{c : [y] \to [c] \text{ is in G and } c \in C\}$.

4 - a directed edge $\xi_x \to [x]$ for each element $x \in E$, where ξ_x is an atom of the dual specifically created for x.

After the transitive closure of the graph G is calculated, step 3 must be reviewed until no more edges are added and the graph is transitively closed. We refer as **"the dual"** to the semilattice D atomized by the atoms $\{\xi_x : x \in E\}$ with upper constant segments given by G.

D.12 Theorems

Theorem 25. *The trace satisfies the following properties:*
i) $Tr(\phi \bigtriangledown \psi) = Tr(\phi) \cup Tr(\psi)$ *for any two atoms* ϕ *and* ψ.
ii) $Tr(r \odot s) = Tr(r) \cap Tr(s)$ *for any two terms* r *and* s,
iii) $Tr(t) = \cap_{c \in C(t)} Tr(c)$ *for any term* t.

Proof. (i) The upper constant segment $U^c(\phi \bigtriangledown \psi) = \{c < \phi \bigtriangledown \psi : c \in C\}$ of a union of atoms $\phi \bigtriangledown \psi$ is defined by $U^c(\phi \bigtriangledown \psi) = U^c(\phi) \cup U^c(\psi)$. Notice that this is a universal property, i.e.

it does not assume the atoms belong to any model. From the definition of the trace of an atom $Tr(\phi) = \{\xi : \exists c \in C,\ (\phi < c) \wedge (D_i \models \xi < [c])\}$, follows that the trace of a union of atoms is $Tr(\phi \bigtriangledown \psi) = \{\xi : \exists c \in C,\ (\phi < c) \vee (\psi < c) \wedge (D_i \models \xi < [c])\} = Tr(\phi) \cup Tr(\psi)$.
(ii) Let M be any atomized model. $Tr(r \odot s) = \cap_{\varphi \in L^a_M(r \odot s)} Tr(\varphi)$ and using $L^a_M(r \odot s) = L^a_M(r) \cup L^a_M(s)$ then $Tr(r \odot s) = (\cap_{\varphi \in L^a_M(r)} Tr(\varphi)) \cap (\cap_{\varphi \in L^a_M(s)} Tr(\varphi)) = Tr(r) \cap Tr(s)$.
(iii) $Tr(t) = \cap_{\phi \in L^a_M(t)} Tr(\phi) = \cap_{c \in C(t)} \cap_{\phi \in L^a_M(c)} Tr(\phi) = \cap_{c \in C(t)} Tr(c)$ where we have used $L^a_M(t) = \cup_{c \in C(t)} L^a_M(c)$. □

Theorem 26. *Let ϕ be any atom and t any term:*

$$\cap_{\psi_k \in L^a_M(t)} Tr(\phi \bigtriangledown \psi_k) = Tr(\phi) \cup Tr(t).$$

Proof. This is a consequence of the linearity of the trace (see Theorem 25) for the atoms;

$$\cap_{\psi_k \in L^a_M(t)} Tr(\phi \bigtriangledown \psi_k) = \cap_{\psi_k \in L^a_M(t)} (Tr(\phi) \cup Tr(\psi_k)) = Tr(\phi) \cup (\cap_{\psi_k \in L^a_M(t)} Tr(\psi_k))$$

and using $\cap_{\psi_k \in L^a_M(t)} Tr(\psi_k) = Tr(t)$ we get to $Tr(\phi) \cup Tr(t)$. □

Theorem 27. *Let $\phi \in dis_M(r)$ in a model M. If M satisfies the positive trace constraint $r \in R_i^+ \rightarrow Tr(r_R) \subseteq Tr(r_L)$, then $Tr(\phi) = \cap_{\psi_k \in L^a_M(r_R)} Tr(\phi \bigtriangledown \psi_k)$.*

Proof. According to Theorem 26, $\cap_{\psi_k \in L^a_M(r_R)} Tr(\phi \bigtriangledown \psi_k) = Tr(\phi) \cup Tr(r_R)$. Assume the positive trace constraint holds. Then $Tr(r_R) \subseteq Tr(r_L) = \cap_{\varphi \in L^a_M(r_L)} Tr(\varphi)$ where the definition for the trace of a regular element $Tr(x) = \cap_{\varphi \in L^a_M(x)} Tr(\varphi)$ has been used. Since $\phi \in dis_M(r)$ implies $\phi \in L^a_M(r_L)$ it follows that $Tr(r_L) \subseteq Tr(\phi)$. Therefore, $Tr(r_R) \subseteq Tr(r_L) \subseteq Tr(\phi)$. Hence, $Tr(\phi) \cup Tr(r_R) = Tr(\phi)$, and also $\cap_{\psi_k \in L^a_M(r_R)} Tr(\phi \bigtriangledown \psi_k) = Tr(\phi)$. □

Theorem 28. *Let $\phi \in dis_M(r)$. The replacement $\phi \rightarrow \{\phi \bigtriangledown \psi_k : \psi_k \in S_\phi \subseteq L^a_M(r_R)\}$ where S_ϕ is a subset of $L^a_M(r_R)$ that satisfies $Tr(\phi) = \cap_{\psi_k \in S_\phi} Tr(\phi \bigtriangledown \psi_k)$ is a trace invariant operation, i.e. it keeps the trace of all atoms and regular elements of M unchanged. If M satisfies the positive trace constraint for $r \in R_i^+ \rightarrow\ Tr(r_R) \subseteq Tr(r_L)$ such set S_ϕ always exists.*

Proof. Let N be the model resulting from replacing ϕ in M. Traces of atoms depend upon the dual only and not upon the presence of other atoms in M, therefore their traces remain unchanged.

Let x be a regular element and assume $\psi_k < x$ and $\phi \not< x$ for some k. If x is in the upper segment of ψ_k, then some component constant c of x is in the upper segment of ψ_k and it follows that c is also in the upper segment of $\phi \bigtriangledown \psi_k$. Therefore, any regular element x that satisfies $\psi_k \in L^a_M(x)$ in M satisfies $\{\psi_k, \phi \bigtriangledown \psi_k\} \subseteq L^a_N(x)$ in N. However, because $Tr(\psi_k) \subseteq Tr(\phi \bigtriangledown \psi_k)$, which is a consequence of Theorem 25, the new atoms $\phi \bigtriangledown \psi_k$ have no effect over the trace of x; indeed, $Tr_N(x) = Tr_M(x) \cap Tr(\phi \bigtriangledown \psi_k)$ and, since $Tr_M(x) \subseteq Tr(\psi_k)$, then $Tr_N(x) = Tr_M(x)$.
Assume now that $\phi < x$. If x is in the upper segment of ϕ then some constant c of x is in the upper segment of ϕ and it follows $\phi \bigtriangledown \psi_k < c$ for every ψ_k because c is also in the upper segment of $\phi \bigtriangledown \psi_k$. Now, ϕ no longer remains on N because it is replaced. In M the trace of x is $Tr_M(x) = \cap_{\varphi \in L^a_M(x)} Tr(\varphi) = \cap_{\varphi < x : \varphi \in M} Tr(\varphi)$, and in N we have:

$$Tr_N(x) = \cap_{\varphi \in L^a_N(x)} Tr(\varphi) = \cap_{\varphi < x : \varphi \in N} Tr(\varphi) = \cap_{\varphi < x : (\varphi \in M) \wedge (\varphi \neq \phi)} Tr(\varphi) \cap_{\psi_k \in S_\phi} Tr(\phi \bigtriangledown \psi_k) =$$

$$= \cap_{\varphi < x : (\varphi \in M) \wedge (\varphi \neq \phi)} Tr(\varphi) \cap Tr(\phi) = \cap_{\varphi < x : \varphi \in M} Tr(\varphi) = Tr_M(x)$$

where we have used our assumption $Tr(\phi) = \cap_{\psi_k \in S_\phi} Tr(\phi \bigtriangledown \psi_k)$. It follows, $Tr_N(x) = Tr_M(x)$. Since the trace of every atom and every regular element is unaffected by the replacement we conclude that it is a trace invariant operation.

Finally, notice that, according to Theorem 27, if the positive trace constraint is satisfied then $S_\phi = L^a_M(r_R)$ fulfills the required condition, so the set S_ϕ always exists. □

Theorem 29. *For any two regular elements a and b and any model M:*

$$M \models a \leq b \;\Rightarrow Tr(b) \subseteq Tr(a)$$
$$Tr(b) \not\subseteq Tr(a) \;\Rightarrow M \models a \not\leq b,$$

so, if M satisfies the negative trace constraints for R^- then $M \models R^-$.

Proof. $M \models a \leq b$ is equivalent to $L^a_M(a) \subseteq L^a_M(b)$ and the traces $Tr(b) = \cap_{\phi \in L^a_M(b)} Tr(\phi) \subseteq \cap_{\phi \in L^a_M(a)} Tr(\phi) = Tr(a)$. The implication below is just the negation of the first and is the basis for the mechanism that prevents negative duples from becoming positive during the sparse-crossing of the positive duples. □

Theorem 30. *Let D be a dual built for the axioms R. Assume a and b are terms over C that have duals in D;*
i) $D \models [b] \leq [a] \;\Leftrightarrow\; F_C(R^+) \models a \leq b$,
ii) $D \models R^*$ if and only R is consistent.
iii) let ξ_b be the introduced in the graph of the dual with an initial edge to $[b]$. Then,

$$\xi_b \in L^a([a]) \text{ if and only if } \xi_b \text{ exists and } b \in W(a) = \{e : (e \in E) \wedge F_C(R^+) \models a \leq e\}.$$

Remark: only the terms mentioned in R and the pinning terms have a dual, i.e. the elements in the set E of Section D.3. This result does not apply to the remaining $2^{|C|} - 1 - |E|$ terms.

Proof. (i) It is well known that $F_C(R^+) \models (a \leq b)$ if and only if $R^+ \Rightarrow (a \leq b)$ ([6]). Since $R^+ \Rightarrow (a \leq b)$ is equivalent to $R^+ \Rightarrow (b = b \odot a)$, it easily follows that $R^+ \Rightarrow (a \leq b)$ is true if and only if $C(a) \subseteq EC(b)$, where $C(a)$ is the set of component constants of a and $EC(b)$ is an extended set of constants that contains $C(b)$. In fact, $EC(b) = C(b')$ where b' is the term with most component constants that is equal to b modulus R^+. In other words, b' is the largest term that satisfies $R^+ \Rightarrow (b = b')$, which always exists and is unique.
Suppose a constant c such that $F_C(R^+) \models c \leq b = b'$. Since $F_C(R^+)$ is the freest model of R^+, we must have that R^+ implies $c \leq b = b'$. It follows that c must be either in $C(b)$ or in $C(r_L)$ for some duple $r \in R^+$ such that $C(r_R) \subseteq C(b')$. This leads to the equation $EC(b) = C(b) \cup_{r \in R^+ : C(r_R) \subseteq EC(b)} C(r_L)$. We can compute $EC(b)$ iteratively starting with $EC_0(b) := C(b)$ and:

$$EC_n(b) := EC_{n-1}(b) \cup_{r \in R^+ : C(r_R) \subseteq EC_{n-1}(b)} C(r_L),$$

which, since C and R^+ are finite, reaches a limit where $EC_n(b) = EC_{n-1}(b)$ in a finite (usually very small) number of iterations. Indeed, this iterative computation is carried out for each element b with a dual, i.e. for each element of E as part of the procedure to build the graph G that underlies the construction of D; there is an edge $([b] \to [a]) \in G$ if and only if $C(a) \subseteq EC(b)$, which occurs if and only if $F_C(R^+) \models a \leq b$.
(ii) According to Theorem 19, negative duples never imply positive duples. In fact, R implies a positive duple p if and only if R^+ implies p. It follows that the axioms R are inconsistent if and only if there is a duple $s = (s_L, s_R)$ in R^- such that $F_C(R^+) \models s_L \leq s_R$. Since the left and right-hand terms of s are in E, we can use proposition (i) to state that R is inconsistent if and only if $D \models [s_R] \leq [s_L]$. By construction D satisfies the dual of every positive duple $r = (r_L, r_R)$ in R^+, as it is enforced with an edge $([r_R] \to [r_L]) \in G$. Therefore, R is inconsistent if and only if D does not satisfy R^{*-} and the result follows.
(iii) We enforced each positive duple $r \in R^+$ with edges $([r_R] \to [r_L]) \in G$. The atomization of D is constructed by creating an atom ξ_x for each constant $[x]$ of the dual (i.e., for each $x \in E$), by appending the atom to the graph with an edge $\xi_x \to [x]$, and finally by computing the transitive closure of the graph. There is a one-to-one map between the atoms in the lower atomic segment $L^a([d])$ for some $d \in E$ and the set of elements $W(d) = \{e : (e \in E) \wedge D \models [e] \leq [d]\}$. Using (i), there is a one-to-one map between the atoms in the lower atomic segment $L^a([d])$ for some $d \in E$ and the set of elements $W(d) = \{e : (e \in E) \wedge F_C(R^+) \models d \leq e\}$; in fact, the map is given by $\xi_e \in L^a([d])$ if and only if $e \in W(d)$.
The simplification of the atomization of the dual implies removing atoms from the dual with the constraint of having R^{*-} satisfied. In this case some ξ_b atoms may be removed. The removal of atoms does not change the one to one correspondence for the remaining atoms. To extend the validity of (iii), to simplified duals, we must write "$\xi_b \in L^a([a])$ if and only if ξ_b exists and $b \in W(a)$" rather than just "$\xi_b \in L^a([a])$ if and only if $b \in W(a)$". □

Theorem 31. *Let D be a dual built for a set of axioms R and a set E of terms with a dual, as defined in Section D.3. Let $t \in E$ and let ξ be an atom of the dual;*
i) there is a regular element $s \in E$ such that $U^c(\xi) = \{[s]\} \cup U^c([s])$,
ii) if $\xi < [c]$ for every component constant of t then $\xi < [t]$,
iii) $\cap_{c \in C(t)} L^a([c]) = L^a([t])$,
iv) for $t \in E$, there is a duality given by $\cup \leftrightarrow \cap$ and $L^a_{F_C(R^+)}(t) \leftrightarrow L^a_D([t])$.

Proof. (i) An atom ξ in the graph of D is first appended with an edge to a single node, say $[s]$, and only after transitive closure it may gain more edges; let's use ξ_s instead of ξ for more clarity. After transitive closure we may have more edges, like $\xi_s \to [t]$, if either $[s] \to [t]$ or $t = s$. It follows that the upper constant segment of ξ_s is $[s]$ plus the upper segment $U([s])$, i.e. the set of nodes reachable from $[s]$ in the graph of D.
(ii) Let s be the regular element such that $U^c(\xi) = \{[s]\} \cup U^c([s])$, so $\xi_s = \xi$. Assume $\xi_s < [c]$ for every component constant of t. Then, $([s] = [c]) \vee ([s] \to [c])$ for every component constant of t, so $D \models [s] \leq [c]$ for every component constant of t. Immediately follows that $C(t) \subseteq EC(s)$. Since s and t have duals (are elements of E) then, while building the graph G for D, we should have compared t and s and introduced the edge $[s] \to [t]$. After transitive closure we have $\xi_s \to [s] \to [t]$ from which it follows $\xi_s < [t]$. This result can also be proven from Theorem 30 (iii).
(iii) We can rewrite (ii) as $\cap_{c \in C(t)} L^a([c]) \subseteq L^a([t])$. For every component constant c of t there is an edge $[t] \to [c]$ in the graph of D hence, after transitive closure $L^a([t]) \subseteq L^a([c])$, which completes the proof.
Another way to prove this is, again, by using Theorem 30 (iii). In any model $t \leq e$ if and only if $c \leq e$ for each $c \in C(t)$, then $\xi_s \in L^a([t])$ if and only if $s \in \{e : (e \in E) \wedge F_C(R^+) \models t \leq e\}$ if and only if $s \in \cap_{c \in C(t)} \{e : (e \in E) \wedge F_C(R^+) \models c \leq e\}$.
(iv) According to Theorem 30 (i) when $a, b \in E$ we must have $D \models [b] \leq [a] \Leftrightarrow F_C(R^+) \models a \leq b$ and, since D is atomized, $F_C(R^+) \models a \leq b$ if and only if $L^a_D([b]) \subseteq L^a_D([a])$. From proposition iii, in the dual $\cap_{c \in C(t)} L^a_D([c]) = L^a([t])$, while, $\cup_{c \in C(t)} L^a_{F_C(R^+)}(c) = L^a_{F_C(R^+)}(t)$ for $F_C(R^+)$. Therefore, there is a duality, that only applies to terms with a dual, given by $\cup \leftrightarrow \cap$ and $L^a_{F_C(R^+)}(t) \leftrightarrow L^a_D([t])$. □

Theorem 32. *Let $W(t) = \{e : (e \in E) \wedge F_C(R^+) \models t \leq e\}$ be the upper segment restricted to E, introduced in Theorem 30 (iii);*
i) $Tr(\phi) = \cup_{c \in U^c(\phi)} L^a([c])$ for any atom ϕ,
ii) If ξ is an atom of the dual, then $\xi \in Tr(\phi)$ if and only if $s \in \cup_{c \in U^c(\phi)} W(c)$, where $s \in E$ is the regular element associated to ξ of Theorem 31 (i),
iii) $\xi_s \in Tr(t)$ if and only if $s \in \cap_{\phi \in L^a_M(t)} \cup_{c \in U^c(\phi)} W(c)$,
iv) If t has a dual, $L^a([t]) \subseteq Tr(t)$.

Proof. (i) It follows from the definition $Tr(\phi) = \{\xi : \exists c \in C, (\phi < c) \wedge (D \models \xi < [c])\} = \cup_{c \in U^c(\phi)} \{\xi : D \models \xi < [c]\} = \cup_{c \in U^c(\phi)} L^a([c])$ where D is the dual.
(ii) Let ξ_e be the atom corresponding with $[e]$ in Theorem 31 (i):

$$Tr(\phi) := \{\xi : \exists c\ (\phi < c) \wedge (c \in C) \wedge (D \models \xi < [c])\} =$$
$$= \{\xi_e : \exists c, e\ (\phi < c) \wedge (c \in C) \wedge (e \in E) \wedge F_C(R^+) \models c \leq e\} =$$
$$= \cup_{c \in U^c(\phi)} \{\xi_e : (e \in E) \wedge F_C(R^+) \models c \leq e\} = \cup_{c \in U^c(\phi)} \{\xi_e : e \in W(c)\}$$

where we have used Theorem 30 (i) and (iii). From here, it follows that there is a one-to-one map between the trace of an atom ϕ and the intersection with E of the union of the upper segments of the constants $U^c(\phi)$ in the free model of R^+.
(iii) We just showed that $\xi_s \in Tr(\phi)$ if and only if $s \in \cup_{c \in U^c(\phi)} W(c)$. Since $Tr(t) = \cap_{\phi \in L^a_M(t)} Tr(\phi)$, we have that $\xi_s \in Tr(t)$ holds if and only if $s \in \cup_{c \in U^c(\phi)} W(c)$ is true for every atom $\phi \in L^a_M(t)$.
(iv) an atom ξ in the lower segment of $[t]$ is edged to $[t]$ in the graph of D. Since there are edges from $[t]$ to its component constants, by transitive closure ξ is in the lower segment of every component constant of $[t]$. From $Tr(t) = \cap_{c \in C(t)} Tr(c) = \cap_{c \in C(t)} \cap_{\phi \in L^a_M(c)} Tr(\phi)$ and $Tr(\phi) = \{\eta : \exists c\ (\phi < c) \wedge (c \in C) \wedge (D \models \eta < [c])\} = \cup_{\phi < c} \{\eta : D \models \eta < [c]\}$ it follows that $\xi \in Tr(t)$. □

Theorem 33. *Redundant atoms of M do not change the traces in M. The trace is well defined, i.e. the trace is the same for a model independently of how it is atomized.*

Proof. Let x be any term. The trace of x in the model M is given by $Tr(x) = \cap_{\phi \in L^a_M(x)} Tr(\phi)$. Suppose φ and ψ are two atoms of M. If neither φ nor ψ are in the lower atomic segment of x then the trace of x in the model $M + \{\varphi \bigtriangledown \psi\}$ is the same that the trace in M. Assume that φ is in the lower atomic segment of x. The trace of x in the model $M + \{\varphi \bigtriangledown \psi\}$ becomes $Tr'(x) = Tr(\varphi \bigtriangledown \psi) \cap_{\phi \in L^a_M(x)} Tr(\phi) = (Tr(\varphi) \cup Tr(\psi)) \cap Tr(x) = (Tr(\varphi) \cap Tr(x)) \cup (Tr(\psi) \cap Tr(x)) = Tr(x) \cup (Tr(\psi) \cap Tr(x)) = Tr(x)$. Therefore, adding redundant atoms does not alter the traces. On the other hand, the non-redundant atoms of a model are always present in any atomization of a model. It follows that the traces are determined only by the non-redundant atoms of the model M and, hence, are independent of the atomization of the model. Notice that the traces of atoms are universal and do not depend upon M, so we do not need to distinguish $Tr'(\phi)$ and $Tr(\phi)$ as they are equal. □

Theorem 34. *The freest semilattice model, $F_C(\emptyset)$, satisfies the trace constraints of any consistent set of axioms.*

Proof. Theorem 13 in Appendix A.2 shows that the freest model generated by a set of constants C can be atomized by a set A with as many atoms as constants in C, each with a single (and different) constant in its upper constant segment. Let D be a dual built for a consistent set of axioms R, and a set of elements E. Let t be a regular element (a term) with a dual and let ϕ_c be the atom of $F_C(\emptyset)$ such that $U^c(\phi_c) = \{c\}$. Using Theorem 33, we can compute the trace of an element of $F_C(\emptyset)$ assuming that $F_C(\emptyset)$ is atomized by A. For any constant $c \in C$, in the freest model $Tr(c) = \cap_{\varphi \in L_{F_C(\emptyset)}(c)} Tr(\varphi) = \cap_{\varphi < c, \varphi \in A} Tr(\varphi) = Tr(\phi_c)$. Using Theorem 32 (i) we get $Tr(\phi_c) = L^a([c])$ and it follows $Tr(c) = L^a([c])$, Therefore, in the freest model, trace and lower segment of the dual coincide for the constants. We will see now that they also coincide for every element.
The trace of a regular element of $F_C(\emptyset)$ represented by a term t is $Tr(t) = \cap_{\varphi \in L^a_{F_C(\emptyset)}(t)} Tr(\varphi) = \cap_{\varphi \in L^a_A(t)} Tr(\varphi) = \cap_{c \in C(t)} Tr(\varphi_c) = \cap_{c \in C(t)} L^a([c])$. Now using that $\cap_{c \in C(t)} L^a([c]) = L^a([t])$ was proven in Theorem 31 (iii), we can write $Tr(t) = \cap_{c \in C(t)} L^a([c]) = L^a([t])$. It follows that, for the freest model $F_C(\emptyset)$ the equality $Tr(t) = L^a([t])$ holds for every element of E. Therefore, for $F_C(\emptyset)$, the trace constraints $Tr(r_R) \subseteq Tr(r_L)$ for $r \in R^+$ and $Tr(r_R) \not\subseteq Tr(r_L)$ for $r \in R^-$ become $L^a([r_R]) \subseteq L^a([r_L])$ for $r \in R^+$ and $L^a([r_R]) \not\subseteq L^a([r_L])$ for $r \in R^-$, which correspond to axioms in the set R^* that axiomatizes D, so they are always satisfied by the dual if R is consistent, as proven in Theorem 30. □

Theorem 35. *Given a model M and a dual D built for a consistent set of axioms R, there is a set Λ of atoms with a single constant in their upper constant segments such that $M + \Lambda$ satisfy the trace constraints. The set Λ can be obtained with Algorithm 5.*

Proof. The first part can be proven by making $\Lambda = \{\phi_c : U^c(\phi_c) = \{c\} \wedge c \in C\}$; since every atom over C is redundant with such set Λ, the atomization $M + \Lambda$ atomizes $F_C(\emptyset)$ and the result follows by using Theorem 34. However, since trace constraints can usually be fixed with a set Λ of a carnality significantly smaller than $|C|$, we provide a proof here more in line of what is needed to actually compute Λ.

In Algorithm 5 we start with $\Lambda = \emptyset$ and add atoms to Λ as needed. Let $N = M + \Lambda$ at some intermediate step.

Let $r \in R^+$ and assume the trace constraint for r is "broken", i.e. $Tr_N(r_R) \not\subseteq Tr_N(r_L)$. There must be at least one atom ξ of D such that $\xi \in Tr_N(r_R) = \cap_{\varphi \in L^a_N(r_R)} Tr(\varphi)$ and $\xi \notin Tr_N(r_L) = \cap_{\varphi \in L^a_N(r_L)} Tr(\varphi)$. Suppose that $\xi < [c]$ for every constant c in the component constants of r_R. Theorem 31 establishes that $\xi < [r_R]$ and, since r is in the set of positive axioms, the graph for D has an edge $[r_R] \to [r_L]$; it follows $\xi < [r_L]$ and, since there are edges from $[r_L]$ to the duals of the component constants of r_L, every atom $\varphi \in L^a_M(r_L)$ must satisfy $\xi \in Tr(\varphi)$ which is clear from the

definition $Tr(\varphi) = \{\xi : \exists c\ (\varphi < c) \wedge (c \in C) \wedge (D \models \xi < [c])\}$. This implies $\xi \in Tr_N(r_L) = \cap_{\varphi \in L^a_N(r_L)} Tr(\varphi)$; a contradiction. Therefore, there is a component constant c of r_R for which $\xi \not< [c]$. The atom ϕ_c with $U^c(\phi_c) = \{c\}$ has a trace $Tr(\phi_c) = L^a([c])$ that does not contain ξ. The model $N + \{\phi_c\}$ satisfies $\xi \notin Tr_{N+\{\phi_c\}}(r_R) = \cap_{\varphi \in L^a_{N+\{\phi_c\}}(r_R)} Tr(\varphi) = Tr_N(r_R) \cap L^a([c])$. Add ϕ_c to Λ and update $N = M + \Lambda$. Repeating the same procedure for every atom in $Tr_N(r_L) - Tr_N(r_R)$ we can "fix" the positive trace constraint, i.e. we reach a set Λ such that $Tr_{M+\Lambda}(r_R) \subseteq Tr_{M+\Lambda}(r_L)$.

Assume now $r \in R^-$ and the trace constraint for r is broken, i.e. $Tr_N(r_R) \subseteq Tr_N(r_L)$. Since D has been built for R, a consistent set of axioms, it must satisfy $D \models [r_R] \not\leq [r_L]$ from which it follows that there is at least one atom ξ of D such that $\xi < [r_R]$ and $\xi \not< [r_L]$. Since there are edges in the graph of D from r_R to the dual of each component constants c of r_R, the duple $\xi < [r_R]$ implies $\xi < [c]$ for each component constant and then $\xi \in Tr(r_R)$. On the other hand, using Theorem 31 and $\xi \not< [r_L]$ there must be a component constant c of r_L for which $\xi \not< [c]$. The atom ϕ_c with $U^c(\phi_c) = \{c\}$ has a trace $Tr(\phi_c) = L^a([c])$ that does not contain ξ. The model $N + \{\phi_c\}$ satisfies $\xi \notin Tr_{N+\{\phi_c\}}(r_L) = \cap_{\varphi \in L^a_{N+\{\phi_c\}}(r_L)} Tr(\varphi) = Tr_N(r_L) \cap L^a([c])$, and the negative trace constraint is fixed, i.e. $Tr_{N+\{\phi_c\}}(r_R) \not\subseteq Tr_{N+\{\phi_c\}}(r_L)$. Add ϕ_c to Λ and update N.

Since adding atoms to a model alters its traces, fixing one trace constraint with an atom implies that the whole set of trace constraints must be reviewed. If a trace constraint breaks by the insertion of an atom in Λ we can always fix it by adding more atoms to Λ. This process necessarily ends as there are at most $|C|$ atoms that can added. In the worst case, Λ contains the $|C|$ atoms with a single constant in its upper constant segment, and $M + \Lambda$ becomes the freest model, i.e. $F_C(\emptyset)$, which satisfies the trace constraints for any dual as discussed above. □

Theorem 36. *Let D be a dual defined for a set of axioms R. Let M be a model that satisfies the positive trace constraints for R^+. Let a and b be any two terms with or without a dual;*
i) If $M \models a \leq b$ then the traces of M satisfy $Tr_M(b) \subseteq Tr_M(a)$,
ii) if $F_C(R^+) \models a \leq b$ then the traces of M satisfy $Tr_M(b) \subseteq Tr_M(a)$,
iii) if $\Box_{R^+} M \models a \leq b$ then the traces of M satisfy $Tr_M(b) \subseteq Tr_M(a)$.

Proof. (i) This was proven in Theorem 29.
(ii) In the proof of Theorem 30 we used that $F_C(R^+) \models a \leq b$ if and only if $R^+ \Rightarrow a \leq b$ simply because $F_C(R^+)$ is the free model of R^+. We also showed that there is a term b' such that $F_C(R^+) \models (b = b')$ that has the property $F_C(R^+) \models a \leq b$ if and only if $C(a) \subseteq C(b')$. The term b' is the one with most component constants in the class of terms equivalent to b in $F_C(R^+)$. In fact, b' is the only term equivalent to b such that for every $r \in R^+$ and $C(r_R) \subseteq C(b')$ then $C(r_L) \subseteq C(b')$. We argued in the proof of Theorem 30 that we can construct b' starting from $b_0 = b$ and extending b to $b_{n+1} = b_n \odot r_L$ if $C(r_R) \subseteq C(b_n)$ for $r \in R^+$, obtaining a series of terms that converges to b' in a finite number of steps. From Theorem 25 (iii), it is clear that $C(r_R) \subseteq C(b_n)$ implies $Tr(b_n) \subseteq Tr(r_R)$. Assume the positive trace constraints are satisfied. We have $Tr_M(b_n) \subseteq Tr_M(r_R) \subseteq Tr_M(r_L)$ and then $Tr_M(b_{n+1}) = Tr_M(b_n) \cap Tr_M(r_L) = Tr_M(b_n)$. It follows that $Tr_M(b') = Tr_M(b)$, in other words, b converges towards b' in a series of terms with increasingly more constants but always with the same trace. Even more, since every term t in the set $\{t : F_C(R^+) \models (b = t)\}$ can be extended to b' we have that $F_C(R^+) \models (b = t)$ implies $Tr_M(b) = Tr_M(t)$. Using this result and the fact that $a \leq b$ is equivalent to $b = a \odot b$, we have $F_C(R^+) \models a \leq b$ implies $Tr_M(b) = Tr_M(a \odot b)$. From $Tr_M(a \odot b) = Tr(a) \cap Tr(b)$ (Theorem 25 (ii)) it follows that $Tr_M(b) \subseteq Tr_M(a)$.
(iii) The model $\Box_{R^+} M$ is the freest model $F_C(Th^+(M) \cup R^+)$. We can follow the same reasoning that we applied to $F_C(R^+)$ in (ii). We have $F_C(Th^+(M) \cup R^+) \models a \leq b$ if and only if $Th^+(M) \cup R^+ \Rightarrow a \leq b$. Again, there must be some b' such that $C(a) \subseteq C(b')$ and we could construct such b' using an iterative process with the duples of $Th^+(M) \cup R^+$ instead of just R^+. The term $b_0 = b$ grows into b' in a series of steps for which $Tr_M(b_{n+1}) = Tr_M(b_n) \cap Tr_M(r_L)$ for some $r \in Th^+(M) \cup R^+$. We showed in (i) and (ii) that for every $r \in Th^+(M) \cup R^+$ the traces of M satisfy $Tr_M(r_R) \subseteq Tr_M(r_L)$. As before, it follows $Tr_M(b_{n+1}) = Tr_M(b_n) \cap Tr_M(r_L) = Tr_M(b_n)$ and then, $Th^+(M) \cup R^+ \Rightarrow a \leq b$ implies $Tr_M(b) = Tr_M(b)' \subseteq Tr_M(a)$. Therefore, if $\Box_{R^+} M \models a \leq b$ the traces of M satisfy $Tr_M(b) \subseteq Tr_M(a)$. □

Theorem 37. *Let D be a dual built for a set E of terms and axioms R. Let $T_\psi \in E$ be the pining term of an atom ψ and $c \in C \subset E$ a constant such that $c \notin C(T_\psi)$. If $D \models [T_\psi] \leq [c]$ then ψ is not an atom of $F_C(R^+)$.*

Proof. If $D \models [T_\psi] \leq [c]$ and $c \notin C(T_\psi)$ the dual must have acquired an edge $[T_\psi] \to [c]$ as a consequence of the enforcing of R^{*+}. There should be some mentioned terms $s_1, s_2, ...s_n$ (therefore with a dual) such that $[T_\psi] \to [s_1] \to ... \to [s_n] \to [c]$ by which $[T_\psi]$ has acquired the edge $[T_\psi] \to [c]$ from transitive closure. Since $T_\psi, s_i, c \in E$, from Theorem 30 (i), $F_C(R^+) \models c \leq s_n \leq ... \leq s_1 \leq T_\psi$. On the other hand, c is not a component constant of T_ψ, therefore $\psi < c$. From $\psi < c$ and $F_C(R^+) \models c \leq T_\psi$ we conclude that ψ is not an atom of the model $F_C(R^+)$ because $\psi < T_\psi$ is false. □

Theorem 38. *Let ϕ be any atom, D a dual built for a set of axioms R and ξ an atom of the dual. Consider the correspondence for atoms of the dual of Theorem 31 (i):*
i) if ξ corresponds with the pinning term T_ψ of an atom ψ, i.e. if ξ was introduced in the dual with an edge $\xi \to [T_\psi]$, then $\xi \notin Tr(\phi)$ implies $U^c(\phi) \subseteq U^c(\psi)$.
ii) if ξ corresponds with the pinning term of ψ and $\psi \models R^+$ then $\xi \notin Tr(\phi)$ if and only if $U^c(\phi) \subseteq U^c(\psi)$.
iii) if ξ corresponds with the right-hand side of a duple $r \in R^-$, i.e. ξ was introduced with an edge $\xi \to [r_R]$, then $\xi \notin Tr(\phi)$ implies $\phi \not< r_R$.

Proof. (i) Suppose $U^c(\phi) \not\subseteq U^c(\psi)$. There is a constant k such that $\phi < k$ and $\psi \not< k$. The pinning term T_ψ has component constants $C - U^c(\psi)$, so $k \in C(T_\psi)$. In D there is an edge $\xi \to [T_\psi]$ and edges $[T_\psi] \to [c]$ for each constant $c \in C - U^c(\psi)$, including k. Therefore, $D \models \xi < [k]$. Since $Tr(\phi) = \{\eta : \exists c\ (\phi < c) \wedge (c \in C) \wedge (D \models \eta < [c])\} = \cup_{\phi<c}\{\eta : D \models \eta < [c]\}$, it follows $\xi \in Tr(\phi)$. Hence, $\xi \notin Tr(\phi)$ implies $U^c(\phi) \subseteq U^c(\psi)$.
(ii) Left to right is proven in (i). Assume $U^c(\phi) \subseteq U^c(\psi)$. Since $C(T_\psi) = C - U^c(\psi)$, we get $C(T_\psi) \cap U^c(\phi) = \emptyset$. Suppose $\xi \in Tr(\phi) = \cup_{\phi<c}\{\eta : D \models \eta < [c]\}$; there must be at least one constant $c \in U^c(\phi)$ such that $D \models \xi < [c]$, that is not a component constant of T_ψ (because $C(T_\psi) \cap U^c(\phi) = \emptyset$), such that the dual has acquired an edge $[T_\psi] \to [c]$ as a consequence of the enforcing of R^{*+}. Theorem 37 says that in this situation ψ is not an atom of the model $F_C(R^+)$. Hence, if $\psi \models R^+$ then $U^c(\phi) \not\subseteq U^c(\psi)$ implies $\xi \notin Tr(\phi)$.
(iii) In the graph of D there is an edge $[r_R] \to [c]$ for each component constant c of r_R and an edge for the atom $\xi \to [r_R]$, so $D \models \xi < [c]$ for every component constant of r_R. Suppose $\phi < r_R$. There is a constant $k \in U^c(\phi) \cap C(r_R)$. Since $Tr(\phi) = \cup_{\phi<c}\{\eta : D \models \eta < [c]\}$ and $\phi < k$ and $\xi < [r_R] < [k]$ it follows $\xi \in Tr(\phi)$. Hence, if $\xi \notin Tr(\phi)$ then $\phi \not< r_R$. □

Theorem 39. *If a model M over a set of constants C satisfies the axioms $R = R^+ \cup R^-$, there is a model N over C spawned by at most $|R^-| + 1$ atoms of M, such that N also satisfies R.*

Proof. If $M \models R$, any model N atomized with a subset of atoms of M satisfies $N \models R^+$; this is clear, since the discriminant of a duple $r \in R^+$ that is empty in M it is also empty in N. For each duple in $s \in R^-$ choose one atom of M in the discriminant of s: the resulting subset of atoms discriminates every duple in R^-. Since N also satisfies R^+, we have $N \models R$. Not every subset of atoms of M satisfies Axiom AS1 so it does not correspond to a proper atomization. This, however, can easily be solved by appending the null atom $\ominus_C$ (see Definition 18) to the subset.The atom $\ominus_C$ is compatible with every semilattice model over C as it discriminates no duple so, in practice, it can be appended to any model and it can also be ignored while doing computations. It follows that a subset of at most $|R^-| + 1$ atoms of M atomizes a model over C that satisfies R. □

Theorem 40. *Let D be a dual built for axioms R and atomized by a set of atoms $D_N \cup D_R$ where atoms in D_N correspond to pinning terms, and atoms in D_R correspond to right-hand side terms of duples of R^-, according to the correspondence of Theorem 31 (i). Let M be a model that satisfies the trace constraints for the dual D:*
i) There is a subset of atoms of M, of at most $n \leq \sum_{\xi_\psi \in D_N} |U^c(\psi)| + \sum_{\xi_{r_R} \in D_R} |C - C(r_R)|$.

atoms, that atomizes a model that also satisfies the trace constraints. This subset can be obtained by discarding atoms of M while keeping the traces of the constants invariant.
ii) If the atomization of the dual is dominated by the pinning terms, i.e, when $|D_R| << |D_N|$, an upper bound for n is $n \leq \alpha|D|$ where α is the average size of the upper constant segment of the atoms in the set $\{\psi : \xi_\psi \in D_N\}$ and $|D| \approx |D_N|$ is the size of the atomization of the dual.

Proof. The dual D is built for a set $E = C \cup Terms(R^+ \cup R^-) \cup \{T_\psi : \psi \in N\}$ where N is a set of atoms and T_ψ the pinning term of $\psi \in N$. The atoms of the dual are introduced either edged to the dual of a pinning term $[T_\psi]$ or under the dual $[r_R]$ of a duple $r \in R^-$. Since the atoms of the dual associated to pinning terms tend to discriminate many duples of R^{*-}, a dual with pinning terms can often be atomized by a handful of atoms, mostly under pinning terms. In fact, Theorem 39 says that to atomize the dual we need no more than $|R^{*-}| + 1 = |R^-| + 1$ atoms. The result we are going to prove is valid no matter if we chose to reduce the size of the atomization of the dual or if we choose to keep all of the atoms, i.e., as many as $|R^-| + |N|$.
(i) The trace is a function that maps a term to a subset of the atoms of the dual. For any term x the trace in M is $Tr(x) = \cap_{c \in C(x)} Tr(c)$. Hence, a discarding of atoms from M that preserves the traces of the constants preserves the traces of all the terms and then, it also preserves the trace constraints. Since, for a constant c the trace in M is $Tr(c) = \cap_{\phi \in L^a_M(c)} Tr(\phi)$, discarding atoms while preserving the trace $Tr(c)$ requires keeping, for each atom $\xi \in D - Tr(c)$, at least one atom $\phi \in L^a_M(c)$ such that $\xi \notin Tr(\phi)$. Since $\xi \notin Tr(c)$ implies $D \models [\xi] \not\leq [c]$, to preserve the trace of c we need at most $|\{\xi \in D : D \models [\xi] \not\leq [c]\}|$ atoms of M. In total, the number n of atoms of M that we must keep is bounded by

$$n \leq \sum_{c \in C} |\{\xi \in D : D \models [\xi] \not\leq [c]\}| = \sum_{\xi \in D} |\{c : D \models [\xi] \not\leq [c]\}|.$$

We can separate $\sum_{\xi \in D} |\{c : D \models [\xi] \not\leq [c]\}|$ into two summands, one for atoms of the dual associated to pinning terms and another for atoms of the dual associated to the right-hand side of the duples of R^-, i.e., $n \leq \sum_{\xi_\psi \in D_N} |\{c : D \models [T_\psi] \not\leq [c]\}| + \sum_{\xi_{r_R} \in D_R} |\{c : D \models [r_R] \not\leq [c]\}|$. Since there are edges from the dual of a term to its component constants we have the following two set inclusions: $\{c : D \models [T_\psi] \not\leq [c]\} \subseteq U^c(\psi)$ and $\{c : D \models [r_R] \not\leq [c]\} \subseteq C - C(r_R)$, so:

$$n \leq \sum_{\xi_\psi \in D_N} |\{c : D \models [T_\psi] \not\leq [c]\}| + \sum_{\xi_{r_R} \in D_R} |\{c : D \models [r_R] \not\leq [c]\}|$$

$$\lesssim \sum_{\xi_\psi \in D_N} |U^c(\psi)| + \sum_{\xi_{r_R} \in D_R} |C - C(r_R)|.$$

We use $\lesssim$ because the two set inclusions are, approximately, set equalities. According to Theorem 37 if the atoms of N satisfy R^+, which is the case as atoms that do not satisfy R^+ are discarded, then $|\{c : D \models [T_\psi] \not\leq [c]\}| = |U^c(\psi)|$. On the other hand, $|\{c : D \models [r_R] \not\leq [c]\}| \approx |C - C(r_R)|$ in most cases.
(ii) As training progresses it becomes possible (by discarding atoms of the dual as in the proof of Theorem 39) to obtain an atomization of the dual dominated by the pinning terms, $|D| \approx |D_N|$. When the atomization of the dual becomes dominated by the pinning terms we can neglect the contribution of $\sum_{\xi_{r_R} \in D_R} |C - C(r_R)|$ and then $n \leq \sum_{\xi_\psi \in D_N} |U^c(\psi)| \approx \alpha|D|$ where α is the average atom size $\alpha = \frac{1}{|D_N|} \sum_{\xi_\psi \in D_N} |U^c(\psi)|$. □

D.13 Algorithms

Algorithm 1: batch training

Initialize $M_0 = F_C(\emptyset)$ if M_0 is not provided;
$N_0 \leftarrow M_0$;
create m batches $R_i \subset R$, with or without repeating duples;
foreach $i = 1$ *to* m **do**
    $M_i \leftarrow M_{i-1}$;
    $N_i \leftarrow \phi \in N_{i-1}$ such that $\forall r \in R_i^+ \ \phi \notin D(r)$;
    $D \leftarrow$ build the dual for N_i and R_i;
    $\Lambda \leftarrow$ enforce trace constraints for R_i in model M_i with dual D;
    $M_i \leftarrow M_i \cup \Lambda$;
    $M_i \leftarrow$ sparse-crossing of R_i^+ in model M_i with dual D;
    $N_i \leftarrow N_i \cup M_i$;

Algorithm 2: trace of atom ϕ for dual $D = (v,\ L^a([\,]))$

$tr(\phi) = \emptyset$;
foreach $c \in U^c(\phi)$ **do**
    $tr(\phi) \leftarrow tr(\phi) \cup L^a([c])$;
return $tr(\phi)$;

Algorithm 3: trace of term t in model M with dual $D = (v,\ L^a([\,]))$

$tr(t) = \{positions\ of\ v\}$;
foreach $\phi \in M$ **do**
    if $C(t) \cap U^c(\phi) \neq \emptyset$ **then**
        $tr(\phi) \leftarrow$ trace of atom ϕ for dual D;
        $tr(t) \leftarrow tr(t) \cap tr(\phi)$;
return $tr(t)$;

Algorithm 4: build the dual for the set of atoms N and duples R

$I = \emptyset$;
foreach $(r_L, r_R) \in R^-$ **do**
  $ind(r_R) \leftarrow C(r_R)$;
  $I \leftarrow I \cup \{ind(r_R)\}$;
foreach $ind \in I$ **do**
  do
    foreach $(r_L, r_R) \in R^+$ **do**
      if $r_R \subset ind$ **then**
        $ind \leftarrow ind \cup \{component\ constants\ of\ r_L\}$;
  while ind *changes*;
$v \leftarrow$ transform I into a vector without repeated elements. Concatenate N at the end;
$T \leftarrow \{r_L : (r_L, r_R) \in R^-\} \cup \{r_R : (r_L, r_R) \in R^-\} \cup C$;
foreach $t \in T$ **do**
  $L^a([t]) = \emptyset$;
  foreach $k \in v$ **do**
    if $k \in I$ **then**
      if $t \subseteq k$ **then**
        $L^a([t]) \leftarrow L^a([t]) \cup \{position\ of\ k\ in\ v\}$;
    if $k \in N$ **then**
      if $t \cap U^c(k) = \emptyset$ **then**
        $L^a([t]) \leftarrow L^a([t]) \cup \{position\ of\ k\ in\ v\}$;
foreach $(r_L, r_R) \in R^-$ **do**
  if $L^a([r_H]) \subseteq L^a([r_L])$ **then**
    return "R is inconsistent";
optional: discard one or many elements of v if $\forall (r_L, r_R) \in R^-$ still holds $L^a([r_H]) \not\subseteq L^a([r_L])$;
return v and $\{L^a([c]) : c \in C\}$;

Algorithm 5: enforce trace constraints for R in model M and dual $D = (v,\ L^a([\,]))$

foreach $\phi \in M$ **do**
  $tr(\phi) \leftarrow$ trace of atom ϕ for dual D;
$T \leftarrow \{r_L : (r_L, r_R) \in R\} \cup \{r_R : (r_L, r_R) \in R\} \cup C$;
foreach $t \in T$ **do**
  $tr(t) \leftarrow$ trace of term t in M with dual D;
$\Lambda \leftarrow \emptyset$;
do
  foreach $(r_L, r_R) \in R^-$ **do**
    if $tr(t_R) \subseteq tr(r_L)$ **then**
      $dC \leftarrow C(r_L) \backslash C(r_R)$;
      do
        $c \leftarrow$ randomly extract c from dC with removal;
        if $tr(r_R) \backslash \big(tr(r_L) \cap L^a([c])\big) \neq \emptyset$ **then**
          $\phi \leftarrow$ create atom such that $U^c(\phi) = \{c\}$;
          $tr(\phi) \leftarrow L^a([c]))$;
          $\Lambda \leftarrow \Lambda \cup \{\phi\}$;
          foreach $t \in T$ **do**
            if $c \in C(t)$ **then**
              $tr(t) \leftarrow tr(t) \cap L^a([c]))$;
          $dC = \emptyset$;
      while $dC \neq \emptyset$;
  foreach $(r_L, r_R) \in R^+$ **do**
    if $tr(t_R) \not\subseteq tr(r_L)$ **then**
      $dI = tr(t_R) \backslash tr(r_L)$; $dC \leftarrow C(r_R) \backslash C(r_L)$;
      do
        $c \leftarrow$ randomly extract c from dC with removal;
        if $dI \cap tr(c) \neq dI$ **then**
          $dI \leftarrow dI \cap tr(c)$;
          $\phi \leftarrow$ create atom such that $U^c(\phi) = \{c\}$;
          $tr(\phi, D) \leftarrow L^a([c])$;
          $\Lambda \leftarrow \Lambda \cup \{\phi\}$;
          foreach $t \in T$ **do**
            if $c \in C(t)$ **then**
              $tr(t) \leftarrow tr(t) \cap L^a([c]))$;
      while $dI \neq \emptyset$;
while Λ *changes*;
return Λ and $\{tr(\phi) : \phi \in M \cup \Lambda\}$ and $\{tr(c) : c \in C\}$;

Algorithm 6: lower atomic segment of term t in model M

$L^a_M(t) \leftarrow \emptyset$;
foreach $\phi \in M$ **do**
  if $C(t) \in U^c(\phi)$ **then**
    $L^a_M(t) \leftarrow L^a_M(t) \cup \{\phi\}$;
return $L^a_M(t)$;

Algorithm 7: Sparse-crossing of R^+ in M with trace $tr(\,)$ and dual $D = (v,\, L^a([\,]))$

$V \leftarrow \{positions\ of\ v\}$;
$S \leftarrow M$;
foreach $(r_L, r_R) \in R^+$ **do**
  $dis \leftarrow L^a_S(r_L) \backslash L^a_S(r_R)$;
  if $dis \neq \emptyset$ **then**
    foreach $i \in V$ **do**
      $tD(i) \leftarrow \emptyset$;
    foreach $\varphi \in L^a_S(r_R)$ **do**
      $dT \leftarrow V \backslash tr(\varphi)$;
      foreach $i \in dT$ **do**
        $tD(i) \leftarrow tD(i) \cup \{\varphi\}$;
    $S \leftarrow S \backslash dis$;
    foreach $\phi \in dis$ **do**
      $dT \leftarrow V \backslash tr(\phi)$;
      do
        $i \leftarrow$ randomly extract index from dT with removal;
        $\varphi \leftarrow$ randomly choose an atom from $tD(i)$;
        $dT \leftarrow dT \cap tr(\varphi)$;
        $\phi \bigtriangledown \varphi \leftarrow$ create atom such that $U^c(\phi) = U^c(\phi) \cup U^c(\varphi)$;
        $tr(\phi \bigtriangledown \varphi) \leftarrow tr(\phi) \cup tr(\varphi)$;
        $S \leftarrow S \cup \{\phi \bigtriangledown \varphi\}$;
      while $dT \neq \emptyset$;
  if $|S|$ *grows beyond some relative or absolute limit* **then**
    $S \leftarrow$ subset of S that preserves the traces of the constants;
return S;

Algorithm 8: find a subset of M that preserves the traces of the constants with trace function $tr(\,)$ and dual $D = (v,\, L^a([\,]))$

$V \leftarrow \{positions\ of\ v\}$;
foreach $i \in V$ **do**
  $tD(i) \leftarrow \emptyset$;
foreach $\varphi \in M$ **do**
  $dT \leftarrow V \backslash tr(\varphi)$;
  foreach $i \in dT$ **do**
    $tD(i) \leftarrow tD(i) \cup \{\varphi\}$;
$N \leftarrow \emptyset$;
foreach $c \in C$ **do**
  $dT \leftarrow V \backslash tr(c)$;
  do
    $i \leftarrow$ randomly extract index from dT;
    if $tD(i) \cap L^a_M(c) \cap N = \emptyset$ **then**
      $\varphi \leftarrow$ randomly choose an atom from $tD(i) \cap L^a_M(c)$;
      $N \leftarrow N \cup \{\varphi\}$;
    else
      $\varphi \leftarrow$ randomly choose an atom from $tD(i) \cap L^a_M(c) \cap N$;
    $dT \leftarrow dT \cap tr(\varphi)$;
  while $dT \neq \emptyset$;
return N;

E Methods

E.1 Computer specifications

For AML computations an AMD Ryzen Threadripper PRO 3975WX 32-Cores with 256 GB of RAM and an AMD Ryzen Threadripper PRO 5965WX 24-Cores with 512 GB of RAM. For statistical learning methods we used an AMD Epyc 7301 @ 2.2 GHz with 128 GB of RAM and a Nvidia GTX 1080 Ti GPU.

E.2 Open-source code and notebooks

An open-source Python/C hybrid implementation of Sparse Crossing is available through the Open AML Engine: https://github.com/Algebraic-AI/Open-AML-Engine.

The dual-language approach used enables researchers to explore and modify the algorithm in Python while maintaining the performance advantages of C. A decorator "@tryfast" in every computationally intensive function provides a way to choose between running the function in Python or in C, facilitating code instrumentation and modification.

The code can compute the Full and Sparse Crossing algorithms. The repository includes example embeddings for various tasks, including Hamiltonian cycle finding, Sudoku, and MNIST handwritten digit classification. The readme includes links to 3 notebooks to guide exploration of three problems: 1) Sparse Crossing of the toy problem that in Section Figure 1 of main text used Full Crossing, 2) Sparse Crossing solution using 200 examples of the MNIST dataset [32], and 3) Sparse Crossing of Hamiltonian cycles

E.3 Datasets and their licences

We used the following image datasets: MNIST [8], FashionMNIST [36], CIFAR-10 [31], and MedMIST medical datasets [32], selected to limit image sizes. We also used the following tabular datsets: bank-marketing [37], Otto-product [38], Amazon employee access [39], Seismic-bumps [40], Gesture Phase Segmentation [33], First-order theorem proving [34], Philippine/Sylvine [41], Phoneme [42], and Satellite [43].

The grayscale datasets from MedMNIST [32] were kept in their original 256-level grayscale depth when analyzed in 28×28 resolution. For medical images in color or in higher resolution, as well as for FashionMNIST and CIFAR-10, to speed up computations, the grayscale or color intensity resolution was reduced from the original 256-level depth to 20 equidistantly distributed levels.

The licenses for the image datasets are:

- CC BY-SA 3.0: MNIST
- CC-BY 4.0: BloodMNIST, OrganCMNIST, PneumoniaMNIST, DermaMNIST
- MIT: Fashion-MNIST, CIFAR-10

The licenses for the tabular datasets are:

- CC BY 4.0: Bank Marketing, Bank Marketing Additional, seismic-bumps, Gesture Phase Segmentation, first order theorem proving
- CC0: Satellite
- Unknown/not specified: Kaggle (Otto Product, Amazon Employee Access), AutoML (Philippine, Sylvine), Other (phoneme, churn)

E.4 Training in Sparse Crossing

Batches. All image datasets were processed following the same protocol. Batch size starts with 500 images and increases linearly until reaching $2/3$ of the training set in batch 500. In the case of tabular datasets, batch size starts with $1/3$ of the entire dataset and linearly grows to cover the entire dataset at batch 666, or $2/3$ of the maximum allowed number of batches, which we set at 1000. Training

stops when the "union model" has 0 error in the training set. A single AML model was obtained for each dataset.

Hyperparameters. Sparse Crossing has 4 hyperparameters. These hyperparameters were set manually and are fixed, i.e., they are not optimized for each individual dataset. The manual setting was carried out based on experience gathered from many synthetic datasets and in MNIST. All other datasets used in this study had no influence on the manual setting of the hyperparameters.

1. *Simplification threshold* γ: during the process of sparse-crossing the positive axioms, if the number of atoms of the master model (see Appendix D.1) grows from a size N to a size larger than γN, a call to a simplification routine triggers. The simplification consists of discarding atoms with the constraint of keeping the traces of all the constants invariant (see Algorithm 8). The simplification parameter has an impact on computation time and it may or may not have an impact on the quality of the models produced, see for example the empirical results for MNIST in Table S1. The value $\gamma = 1.5$ was used for all the image and tabular datasets, while for Hamiltonian cycles the value $\gamma = 1.1$ was set.

2. *Batch size*: The batch size has an impact on computation time and model test accuracy. For image and tabular datasets, we used a policy of making the batch size grow linearly as training progresses, see **Batches**. For Hamiltonian Cycles, all the positive and negative task axioms are presented at each batch.

3. *Union model fractioning parameter* κ: the atoms of the dual are either associated to negative task axioms or to pinning terms. Let D be the set of atoms of the dual, D_N the set of atoms associated to pinning terms and D_R the set of atoms associated to negative task axioms, so $|D| = |D_N| + |D_R|$. The fractioning parameter selects, at random and at each batch, a subset of $S \subseteq D_N$ such that $|D_R| \geq \kappa(|D_R| + |S|)$. In other words, κ is the minimal proportion of atoms associated to negative task axioms that we want in the dual. Since the number of pinning terms increases with training, if this fractioning does not take place, the proportion of atoms associated to axioms decreases. We found that ensuring a proportion κ of atoms associated to axioms helps increase atom variability, i.e. fractioning helps explore a larger volume of the atom space. For image and tabular datasets we used $\kappa = 0.1$ while for Hamiltoinian cycles we observed that larger values, like $\kappa = 0.5$, gave better results. We found this parameter can have a significant impact in model performance for smaller training sets, but in other situations such as the MNIST example the impact is small (Table S1).

4. *Model reduction parameter* δ: Since the accuracy remains approximately constant for a wide range of atomization sizes (see Appendix Figure S6), it is possible to reduce the size of the union model N. To extract a good generalizing model from the union model, a subset of its atoms with size $\delta|N|$ is extracted using the method described in the next section on **Subset selection**. Size reduction with parameter $\delta = 0.1$ was used before the logistic regression and the best-guess discriminant assignment for all datasets. For Hamiltonian cycle problems, no reduction was applied, as each solution is extracted from the master model and not from the union model.

Table S1: Accuracy of best-guess discriminant assignment on a full model trained using 1,000 examples of MNIST, as a function of κ and γ. The values of κ and γ do not have an obvious effect in accuracy, but they do affect computation time and the size of the final model. The used values of κ and γ for image datasets are displayed in bold.

$\kappa \setminus \gamma$	1.3	1.4	**1.5**	1.6	1.7
0.025	89.07%	89.22%	88.77%	88.93%	89.07%
0.05	89.21%	88.99%	88.71%	88.93%	89.22%
0.1	89.10%	89.04%	88.95%	89.14%	89.07%
0.15	89.02%	88.99%	88.95%	89.12%	89.12%
0.2	89.26%	89.14%	89.04%	88.96%	88.85%
0.3	89.14%	89.14%	88.94%	89.05%	89.06%
0.5	89.27%	88.85%	89.03%	89.07%	88.98%

E.5 Wall-clock run-times and computational scaling in Sparse Crossing

We report wall-clock run-times and peak RAM usage at different training-set sizes for a handful of datasets. In particular, regarding image datasets, we report results for Fashion-MNIST (Table S2)

and DermaMNIST (Table S3). And regarding tabular datasets, we report results for churn (Table S4) and Sylvine (Table S5). The results should be interpreted within the current context of AML's implementation, which is in research-stage. The scaling is of interest: wall-clock run-times are about linear to sub-linear, whereas RAM usage is clearly sub-linear. Model size, understood via atom counts, does not follow an obvious pattern particularly for the smallest training set sizes. Run-times from MLPs on image datasets are dominated by the validation step, which is performed on the entire validation set.

Table S2: Training wall-clock run-time and peak RAM usage of AML versus MLPs on Fashion-MNIST, by training set size.

	AML			MLPs (validation on 360 models)	
Size	Train run-time	RAM peak (MB)	Atoms	Train run-time	RAM peak (MB)
50	00:21:30	673	26159	00:11:29	8655
100	00:39:32	825	54713	00:11:51	8535
200	01:15:21	1069	127638	00:12:05	8521
500	02:37:35	1537	351129	00:12:42	8690
1000	05:15:40	2785	727066	00:13:15	8648

Table S3: Training wall-clock run-time and peak RAM usage of AML versus MLPs on DermaMNIST, by training set size.

	AML			MLPs (validation on 360 models)	
Size	Train run-time	RAM peak (MB)	Atoms	Train run-time	RAM peak (MB)
50	00:55:26	1145	16091	00:16:51	8700
100	01:39:59	1443	60708	00:16:08	8673
200	02:47:33	1778	160817	00:16:08	8717
500	06:26:08	3416	481702	00:16:13	8737
1000	12:12:35	5328	1035663	00:16:33	8810

Table S4: Training wall-clock run-time and peak RAM usage of AML versus MLPs on churn, by training set size.

	AML			MLPs (cross-val on 200 models)	
Size	Train run-time	RAM peak (MB)	Atoms	Train run-time	RAM peak (MB)
50	00:03:07	549	1618	00:04:19	19153
100	00:04:06	673	5043	00:02:22	18970
200	00:05:55	550	11027	00:04:07	19338
500	00:12:29	563	33369	00:07:48	19001
1000	00:22:09	768	55011	00:14:01	19500
4500	01:34:32	976	157136	00:51:51	18645

E.6 Subset selection.

Out of the Sparse Crossing procedure we obtain a set of atoms, from which we extract the following subset that we call the AML model. Good generalizing models need subsets of atoms that are individually discriminative, collectively discriminating the entire training set and with low correlation. To build a subset S with these characteristics, we perform several independent computations of subsets of atoms and add the resulting subsets to S, until we reach a number of atoms equal to $1/10$ of the number of atoms in the initial model obtained from Sparse Crossing. In each of these independent computations, we first randomly sort the negative axioms of the training set. Then, beginning with an empty set of atoms A, and for each duple r, we check if there already is an atom in A discriminating r. If not, we randomly select one of the atoms of the Sparse Crossing model that discriminates the

Table S5: Training wall-clock run-time and peak RAM usage of AML versus MLPs on Sylvine, by training set size.

	AML			MLPs (cross-val on 200 models)	
Size	Train run-time	RAM peak (MB)	Atoms	Train run-time	RAM peak (MB)
50	00:03:18	673	2725	00:03:05	18319
100	00:04:14	553	4009	00:04:25	18907
200	00:06:20	569	10846	00:07:01	19469
500	00:11:33	559	22870	00:10:46	19104
1000	00:19:20	586	37633	00:17:16	19069
4611	01:20:24	776	104749	00:56:46	18917

duple and add it to R. The atoms that are not associated to labels (those which upper segment contain no label constants) are removed from the model, as they play no role in associating labels to term images. This protocol results in good generalizing models ten times smaller than the initial union model.

E.7 Logistic regression on top of AML.

If we are interested in adding statistical information to AML, a simple way is to use the AML model as input to logistic regression in the following way. The atoms that are in the lower segment of the image term are given a value of $+1$ and the atoms that are not are given a value of -1. For each example, the input to the logistic regression is then a sequence of $+1$ and -1 values. Each element of the sequence connects with a linear weight to each of N $\mathrm{softmax}$ outputs, one per class. Only the training dataset was used to find optimal parameters, with Adam optimizer [46] and cross-entropy loss [47].

E.8 Methodology for statistical learning comparisons

In all cases, statistical learning methods are trained on the same data used to train AML models, except that in the case of image datasets the statistical learners also use validation data. Below we explain the methodology used for image and tabular datasets.

Methodology for image datasets. Training, validation and test sets were normalized by subtracting the mean and dividing by the standard deviation of the training set. The validation dataset is used to optimize architecture hyperparameters. Hyperparameter selection is done via grid search using Ray Tune [48] in conjunction with Pytorch, with the goal of minimizing validation loss. For all methods, linear layers are initialized using the Kaiming uniform distribution. Training runs for 240 iterations or until the validation loss does not improve for 10 iterations. In each run, we uniformly sample the learning rate ($5 \cdot 10^{-4}$, 10^{-4}, $5 \cdot 10^{-5}$ or 10^{-5}) and the L2-regularization coefficient (10^{-3}, $5 \cdot 10^{-4}$, 10^{-4} or 0) to use as hyperparameters for the ADAM optimizer. The `betas` and `eps` parameters are left as default in Pytorch. We train the models to minimize cross-entropy loss, using a batch size of 128.

For MLPs, we evaluated a family of two, three and four hidden layer multilayer perceptrons with ReLU activations. More concretely, we perform a grid search over the number of neurons in the first hidden layer (512, 2048 or 4096 hidden units) and the second hidden layer (256, 1024 or 2048 hidden units). The third layer, when it exists, is allowed to have 128, 256 or 512 units, and the fourth layer, when it exists, can have 128 or 256 units. For the third and fourth layers, a random sample is performed for the sizes. We perform 90 runs using two layers, 180 runs using three, and an additional 90 runs with four layers, for a total of 360 training runs.

Regarding CNNs, we evaluated a family of two and three convolutional layer networks. More concretely, we perform a grid search over the kernel size of the first convolutional layer (3, 5, 7) and of the second convolutional layer (3, 5, 7). The third convolutional layer, when it exists, is likewise allowed a kernel size of (3, 5, 7). Every convolutional layer uses 32 channels, a padding of 1, and is followed by dropout with probability 0.2 and a ReLU activation. Between consecutive convolutional blocks we insert a max-pooling layer of stride 2, with a pooling window of 3 for kernel size 7 and 2

otherwise. The convolutional stack is followed by a fixed classification head of three fully connected layers with 128 and 64 hidden units respectively, followed by ReLU activations. We perform 90 runs using two convolutional layers and 270 runs using three, for a total of 360 training runs.

Methodology for tabular datasets. Scikit-learn is used to perform a random sweep of hyperparameters using stratified 5-fold cross-validation. Reported test values correspond to test evaluation of a final model trained using the entire training dataset, with hyperparameters corresponding to those that achieved the best mean accuracy between the 5 cross-validation runs. For each dataset, we performed 200 cross-validation runs for MLPs and another 200 for XGBoost, in both cases using the cross-entropy loss.

For MLPs, we use the skorch library to provide compatibility for Pytorch with scikit-learn. Training and test sets were normalized by subtracting the mean and dividing by the standard deviation of the training set, and all models are trained using the Adam optimizer. The cross-validation runs are performed uniformly sampling the learning rate in the interval $(10^{-5}, 10^{-4})$, and with layer sizes uniformly selected from 128, 256, 512, 2048 or 4096 for the first hidden layer, 128, 256, 1024 or 2048 for the second hidden layer, 0, 128, 256 or 512 for the third hidden layer and 0, 128 or 256 for the fourth hidden layer. Layer weights are randomly initialised using the Kaiming uniform distribution. On average we can then expect 50 models to be two layers deep, 100 models to be three layers deep and 50 models to be four layers deep. Intermediate layers use ReLU as the activation function. For each of the 5-fold cross-validation runs, we train for a maximum of 50 epochs, using early stopping, with the accuracy value reported for the fold being the best achieved through the training run. We also report the epoch number at which that accuracy was achieved, and the final model is trained for a number of epochs equivalent to the mean of the number of epochs used between the 5 folds run for the selected hyperparameters. To ensure train samples were balanced between the different classes, the loss was scaled according to the inverse prevalence of each of the classes.

Finally, for XGBoost, we used the `xgboost` library, with parameters randomly sampled as follows: `colsample_bytree` was uniformly sampled from $(0.6, 1)$, `max_depth` was uniformly sampled from 3 to 10, `gamma` was uniformly sampled in $(0, 0.01)$, the learning rate was sampled from a log-uniform in $(10^{-3}, 2 \times 10^{-1})$, `n_estimators` was uniformly sampled between 50 to 500, `reg_alpha` and `reg_lambda` were both uniformly sampled from the interval $(0, 2)$, and `subsample` was uniformly sampled from the interval $(0.6, 1)$. In training, train examples were sampled with a weight inverse to their prevalence in the training set.

E.9 AML in the presence of noise

To empirically observe the behaviour of AML in the presence of noise, we performed training runs of AML on MNIST, either randomly flipping pixels or changing labels. For comparison, we trained MLPs with the same noise as for AML, yet with the advantage of fine-tuning on the noise-free validation set. The results of these experiments are in Table S6 and Table S7, respectively. In both cases, the behaviour of AML accuracies is similar to of MLPs when varying the level of noise. We note that the presence of noise causes the resulting AML model to increase in size: the model needs to encode the rules of the entire training data, including the noise, which necessarily increases the complexity of the rules.

Table S6: Test accuracies observed when training AML on the full binarized MNIST training dataset, where a fixed number of pixels were randomly inverted per training image. For the sake of comparison, MLPs were trained on the same data, following the hyperparameter tuning procedure in Appendix E.8 with an untouched validation set.

Pixels inverted	0		50		100		200	
	Acc.	Atoms	Acc.	Atoms	Acc.	Atoms	Acc.	Atoms
Union model	97.45%	1255067	96.26%	1560572	95.09%	1865441	88.48%	2959155
AML model	97.48%	125506	96.30%	156057	95.04%	186544	88.44%	295915
AML & logistic	98.15%	125506	97.55%	156057	96.58%	186544	88.88%	295915
MLP (360 val)	98.27%		97.72%		97.08%		88.97%	

Table S7: Test accuracies observed when training AML on the full discretised MNIST training dataset, where a fixed rate of examples had their labels randomly reassigned to one of the other labels. For the sake of comparison, MLPs were trained on the same data, following the hyperparameter tuning procedure in Appendix E.8, with an untouched validation set.

Labels flipped	0%		5%		10%		20%	
	Acc.	Atoms	Acc.	Atoms	Acc.	Atoms	Acc.	Atoms
Union model	97.34%	1238763	97.01%	2992466	96.73%	4316733	96.08%	6340802
AML model	97.48%	123876	96.97%	299246	96.87%	431673	96.18%	634080
AML & logistic	98.16%	123876	97.62%	299246	97.22%	431673	96.11%	634080
MLP (360 val)	98.46%		97.26%		97.06%		96.85%	

E.10 Stochasticity in Sparse Crossing

To empirically test the stochasticity of Sparse Crossing, we performed ten independent full training and evaluation runs for MNIST. In accuracy, AML showcases a smaller standard deviation compared to our MLP validation pipeline in all three included modalities, Table S8.

Table S8: Results of 10 repeated training runs on discretised MNIST, using the entirety of the training set. The reported results for MLPs correspond to the full hyperparameter tuning procedure described in Appendix E.8.

Run	Union model		AML model		AML & logistic		MLP
	Acc.	Atoms	Acc.	Atoms	Acc.	Atoms	Acc.
1	97.26%	1250834	97.32%	125083	98.23%	125083	97.97%
2	97.36%	1236773	97.55%	123677	98.25%	123677	97.95%
3	97.38%	1241112	97.62%	124111	98.15%	124111	98.08%
4	97.45%	1255067	97.48%	125506	98.15%	125506	98.27%
5	97.31%	1242656	97.31%	124265	98.27%	124265	98.34%
6	97.35%	1249190	97.48%	124919	98.23%	124919	98.15%
7	97.34%	1238763	97.48%	123876	98.16%	123876	98.46%
8	97.27%	1235217	97.24%	123521	98.22%	123521	96.01%
9	97.11%	1231597	97.11%	123159	98.05%	123159	97.99%
10	97.35%	1243530	97.35%	124353	98.22%	124353	98.18%
Mean±std.	97.32±0.09%		97.39±0.15%		98.19±0.07%		97.94±0.69%

E.11 Formal knowledge ablation

Formal knowledge is incorporated in our analysis of image and tabular datasets via specifying the order relations between the involved numerical variables. To empirically observe what impact the use of this information has, we have tested running our whole pipeline with and without specifying this formal knowledge in two image and two tabular datasets. As a general observation and in the expected way, without formal knowledge the models perform worse and training time also suffers from the added difficulties in finding the rules underlying the data.

For example, when training Fashion-MNIST with 1000 train examples with and without formal knowledge, training time increases from 5:15:40 to 6:13:34, whereas the best-guess discriminant assignment accuracy decreases from 79.72% to 69.99%. And on DermaMNIST, runtime increases from 12:12:35 to 24:59:15, with accuracy decreasing from 68.68% to 66.78% and F1 macro avg. score decreasing from 0.2023 to 0.1144. In fact, the model trained without formal knowledge just predicts every image to be part of the majority class.

In the case of tabular datasets, many values for numerical variables from the test set are not even present in the training set. This seems to be the case in Sylvine, which has 20 numerical features and 0 categorical features. Training time went up from 1:18:25 to 6:35:08, while accuracy fell from 93.76% to 53.99%. Given that this is a binary classification dataset where the most prevalent class corresponds to 51.46% of test examples, this is barely above chance. In churn, which has 20

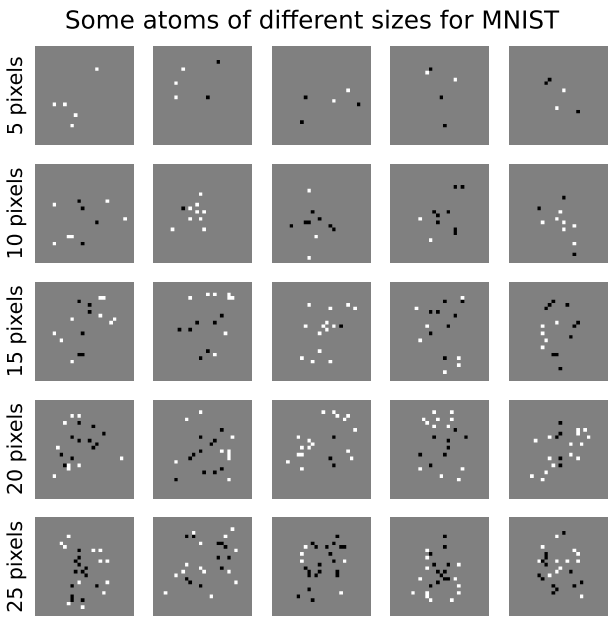


Figure S7: Example representations of atoms with different numbers of constants associated to pixel positions and values for a model trained on MNIST. The constants associated to black or white pixels in a given position are plotted in the position they belong to. If no constant associated to a position in the image is present in the atom we plot it in gray.

numerical features and 4 categorical features, training time increased from 1:36:14 to 2:46:02, and accuracy fell from 90.20% to 85.80%, again barely above chance as the most prevalent class in the test dataset corresponds to 84.4% of the samples.

E.12 Visualization of atoms

For illustrative purposes, in Figure S7 we provide plots of some atoms in different sizes from models trained on MNIST. We observe that many atoms seem to encode the presence of black or white pixels in certain areas of the image. Indeed, same color pixels are often clustered in atoms, or produce rounded shapes which could be expected in the digits.

F Formal problems

As an example of formal problem, we discuss the case of obtaining Hamiltonian cycles.

Encoding the specifications of a Hamiltonian cycle in a semilattice. Consider the following sets of constants:

- V_i: A constant for each graph node
- E_k: A constant for each edge
- P: A constant to refer to the path we want to compute
- W: A constant to encode constraints and allow for training
- nE_k: A constant for the absence of edge k
- Z_k: Auxiliary constants, as many as graph edges
- id_i: a path "id" constant associated to node i
- g_k and h_k: Context constants, as many as graph edges

This gives a total of $2v + 5e + 2$ constants, where v is the number of nodes and e is the number of edges in the graph.

We start by embedding the topology of the graph. Let $r(k)$ and $s(k)$ be the index of the two nodes of edge E_k. The edges are undirected so it does not matter which of the two nodes is $r(k)$ or $s(k)$. For each (undirected) edge k joining nodes $V_{r(k)}$ and $V_{s(k)}$ we define a positive duple:

$$V_{r(k)} \odot V_{s(k)} \leq E_k.$$

The embedding constant Z_k represents either an edge or its absence and is defined with:

$$Z_k \leq E_k \odot nE_k,$$

Think about Z_k as a kind of weak variable that we wish to be equal to either E_k or to nE_k but that can take any value in between. The path P we want to find passes through every node and it is formed with edges, so we add:

$$\odot_i V_i \leq P,$$
$$\odot_k Z_k = P.$$

For the constant W, which we will use to learn the "wrong paths", we start with the following duples; for each edge k it is a wrong path one that simultaneously has the constant of the edge and the constant for the absence of the edge:

$$W \leq E_k \odot nE_k \odot P.$$

Since we want our path not to be a wrong path we also impose the additional negative axiom:

$$W \not\leq P.$$

Then we describe the concept of path with the help of the constants nE_k. For each node i and for each couple of edges E_y and E_z We use:

$$P \odot (\odot_{x;\, x \notin \{y,z\}} nE_x) = E_y \odot E_z \odot P$$

where the idempotent summation $\odot_{x;\, x \notin \{y,z\}} nE_x$ runs along the indexes of every edge of node i, except for y and z. There are a variable number of these positive duples depending upon the graph, on the order of $2v \left(\frac{e}{v}\right)^2$.

We need some negative duples in the embedding (usually, the fewer the better). It is enough with one negative duple for each node i establishing that the presence of node i only depends upon the presence of edges incident to node i and it is independent of everything else:

$$V_i \not\leq (\odot_{j;\, j \neq i} (V_j \odot id_j)) \odot (\odot_k nE_k) \odot (\odot_{t;\, i \notin \{r(t), s(t)\}} E_t)$$

where the idempotent summation $\odot_{j;\, j\neq i}(V_j \odot id_j)$ sums along all the nodes except i, the idempotent summation $\odot_k nE_k$ sums along every edge of the graph, and $\odot_{t;\, i\notin\{r(t),s(t)\}} E_t$ sums along every edge not incident to node i.

To specify we want a connected path, we use the path identity constants id_i. These constants become equal for the nodes in the same path. The identities of adjacent nodes become equal in the presence of a connecting edge:

$$E_k \odot id_{r(k)} = id_{s(k)} \odot E_k.$$

To require that the path connects every two nodes i and j we add:

$$P \odot id_i = id_j \odot P,$$

which gives v^2 duples that are equivalent to just $2v$ duples. To convey the meaning of the node identity we add the following set of negative duples:

$$id_i \not\leq (\odot_{j;\, j\neq i}(V_j \odot id_j)) \odot (\odot_k nE_k) \odot (\odot_{t;\, i\notin\{r(t),s(t)\}} E_t),$$

which has the same right-hand side than the negative duples above.

Using context constants (see [10]) ensures the solution models are spawn by non-redundant atoms and increase the probability of finding a solution; for each edge k we add a context in which Z_k is equal to E_k and another context in which nZ_k is equal to nE_k:

$$g_k \odot Z_k = E_k \odot g_k,$$

$$h_k \odot Z_k = nE_k \odot h_k.$$

The embedding theory has a total of $2v + 1$ negative duples and around $2(e^2/v) + 7e + 2v + 2$ positive duples.

Additionally it is possible to discard paths extracted from the attempts made; if a path is produced that cannot be completed with additional edges, add to the axioms:

$$W \leq \odot_u E_u.$$

where the idempotent summation $\odot_u E_u$ sums along the edges of the unwanted path.

Once a model M of the embedding axioms is produced, we interpret that path P has edge E_k if and only if $E_k \leq P$ is valid in the model M.

Freest model. First, we checked that, for small graphs, we can use the freest model to obtain Hamiltonian cycles. Full Crossing the task axioms for a Petersen graph $(n = 6, k = 2)$ gives $4,253$ atoms, Figure S8a giving the frequencies of atoms and the Petersen graph.

We then selected the atoms that discriminate the task's negative axioms, as we did in the case for learning from data. The resulting discriminant subset has 13 atoms, given in Figure S8a.

Once we have a model using the subset of atoms, we interpret that path P has edge E_k if and only if $(E_k \leq P)$ is positive in the model. For example, we do can check this for the edges $E_{0,1}$ and $E_{0,5}$, for which the discriminants, Eq. 6, are

$$D(E_{0,1} \leq P) = \{\phi_0, \phi_1, \phi_{12}\} \setminus \{\phi_0, \phi_1, ..., \phi_{11}\} = \{\phi_{12}\} \tag{F.1}$$

$$D(E_{0,5} \leq P) = \{\phi_0, \phi_5\} \setminus \{\phi_0, \phi_1, ..., \phi_{11}\} = \{\}. \tag{F.2}$$

$E_{0,1}$ is then discriminated in the model, so $E_{0,1} \not\leq P$ in the model, while $E_{0,1}$ is not discriminated, so $E_{0,1}t \leq P$ in the model. Note that edges constants in atom ϕ_{12} have to correspond to the edges that are not part of the Hamiltonian cycle

$$\{E_{0,1}, E_{0,1}, E_{0,1}, E_{0,1}, E_{0,1}, \}, \tag{F.3}$$

which we leave as black edges in the Petersen graph in Figure S8a, and the remaining edges as red to mark the Hamiltonian cycle. Choosing a different subset of the discriminating atoms gives one the 6 possible Hamiltonian cycles for this graph. Note that the task axioms include no recipe to find the Hamiltonian cycles, in particular no search method, simply the formal description of what something is.

Table S9: Number of attempts needed to obtain a Hamiltonian cycle using Sparse Crossing for graphs 1 to 10 of the FHCP challenge set [35]. Each graph was independently run 10 times. Each run consists of a series of sparse-crossing batches. We make an attempt to find a cycle after each sparse-crossing batch. Each of the 10 runs gives a different number of attempts until obtaining a Hamiltonian cycle. We report the lowest, median and highest number of these 10 values for each graph. Column "SLH Transforms" gives number of graph transformations needed to find a first cycle following the Snakes and Ladders Heuristic algorithm [49], a state of the art algorithm for Hamiltonian cycles.

Graph	Lowest	Median	Highest	SLH Transforms
G1	9	773	3651	13356
G2	12	124	471	5078
G3	808	6798	46419	172316
G4	11	1008	3379	266
G5	1818	10492	64013	81571
G6	28	437	887	370
G7	3560	28202	292521	412275
G8	207	1838	5823	666801
G9	8282	130717		472180
G10	434	768	2907	285

Sparse Crossing. For larger graphs, we used Sparse Crossing to find Hamiltonian cycles in a wide range of random and hard graphs (see Figure S8b and Table S9). We use in the first batch all the duples of the problem, obtain a discriminating subset as we did with Full Crossing, and check if the edges in the path according to the model form a Hamiltonian cycle. If not, we have used one "attempt" and we add as an additional task duple that the path obtained is not a Hamiltonian cycle. We repeat this procedure until a number of attempts that gives a Hamiltonian cycle. We report in Table S9 the number of attempts needed to obtain a Hamiltonian cycle for graphs 1 to 10 of the FHCP challenge set [35]. For each graph, we run Sparse Crossing 10 times, and we report the lowest number of attempts out of the 10 runs, the median number and the maximum number. Some graphs take many more attempts (G3, G5, G7 and G9). We checked that these graphs also take a large number of transformation steps in the Snakes and Ladders Heuristic algorithm [49], a state of the art algorithm for Hamiltonian cycle, see Table S9).

We also need to understand why Sparse Crossing also works in formal problems like this. We note that while we empirically find good models, there exist bad models of the task axioms. For example, in a single attempt we already obey all task axioms yet typically we need more attempts to find a Hamiltonian cycle. The type of encoding we have used here is named as "explicit" embeddings in [10]). That reference demonstrates that each solution of the problem has a model atomized by a subset of non-redundant atoms of the freest model of the task axioms. Since most atoms are redundant, i.e. are unions of non-redundant atoms, this result severely restricts the size of the atom space that contains the solutions. Sparse Crossing is designed to be effective at finding non-redundant atoms (see Appendix D.8), making it look for a solution in the correct portion of the atom space.

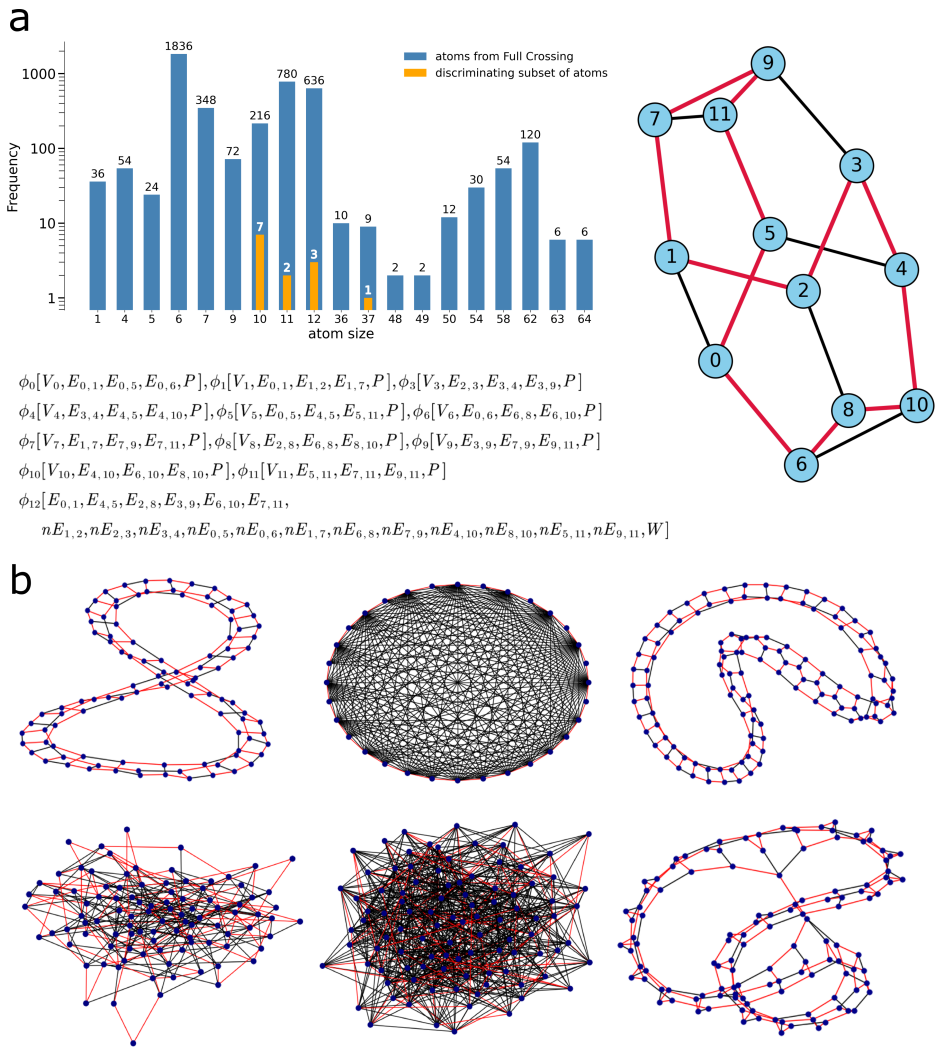


Figure S8: Hamiltonian cycles obtained using AML for different graphs. Sparse Crossing obtains Hamiltonian cycles in graph G7 of the FHCP set [35] (top row, left), Sheehan graphs of various sizes (top row, center), generalized Peterson graphs (GPN_122, top row, right), randomly generated graphs of variable edge density (first two graphs of the bottom row) and modified Flower Snarks (SNm_124, bottom row, right).

G Comparison of accuracies

Table S10: Accuracies and macro average F1 for AML, AML followed by logistic regression, and the best MLP and CNN models selected from a validation dataset across image classification datasets. Below the name of each dataset we indicate the image size, the number of classes and the train/validation/test split. For the best MLPs, we indicate in parentheses the neuron numbers of the hidden layers of the network. For the best CNNs, we indicate in parentheses the kernel sizes of the convolutional layers of the network.

Dataset	AML		AML & logistic regression		MLP best		CNN best	
	accuracy	macro avg. F1	accuracy	macro avg. F1	accuracy	macro avg. F1	accuracy	macro avg. F1
MNIST $28 \times 28, 10,$ 50000/10000/10000	97.63%	0.9761	98.26%	0.9825	98.20% $(4096, 256, 256)$	0.9819	98.26% $(7, 7, 5)$	0.9829
MNIST $28 \times 28, 10,$ 1000/10000/10000	90.24%	0.9011	91.56%	0.9145	88.56% $(2048, 2048)$	0.8840	93.02% $(7, 7, 5)$	0.9296
Fashion-MNIST $28 \times 28, 10,$ 50000/10000/10000	87.27%	0.8715	89.47%	0.8936	89.79% $(4096, 2048)$	0.8976	91.87% $(3, 3)$	0.9188
Fashion-MNIST $28 \times 28, 10,$ 1000/10000/10000	79.72%	0.7960	81.89%	0.8169	81.03% $(2048, 2048, 512, 256)$	0.8106	82.25% $(3, 5, 7)$	0.8231
CIFAR-10 $28 \times 28, 10,$ 50000/5000/5000	48.56%	0.4811	53.58%	0.5306	54.20% $(4096, 1024, 256)$	0.5402	73.26% $(5, 3, 5)$	0.7323
CIFAR-10 $28 \times 28, 10,$ 1000/5000/5000	35.86%	0.3509	37.56%	0.3706	35.46% $(4096, 1024, 128, 256)$	0.3459	44.28% $(3, 7, 5)$	0.4357

Table S10: Accuracies and macro average F1 for AML, AML followed by logistic regression, and the best MLP and CNN models selected from a validation dataset across image classification datasets. Below the name of each dataset we indicate the image size, the number of classes and the train/validation/test split. For the best MLPs, we indicate in parentheses the neuron numbers of the hidden layers of the network. For the best CNNs, we indicate in parentheses the kernel sizes of the convolutional layers of the network (continued).

Dataset	AML		AML & logistic regression		MLP best		CNN best	
	accuracy	macro avg. F1	accuracy	macro avg. F1	accuracy	macro avg. F1	accuracy	macro avg. F1
DermaMNIST $28 \times 28, 7,$ 7007/1003/2005	73.47%	0.3300	74.21%	0.3960	48.08% $(2048, 1024)$	0.2986	67.13% $(7, 5)$	0.4877
PneumoniaMNIST $28 \times 28, 2,$ 4708/524/624	84.62%	0.8205	85.58%	0.8331	84.46% $(512, 1024, 256)$	0.8168	84.13% $(7, 7)$	0.8156
PneumoniaMNIST $64 \times 64, 2,$ 4708/524/624	84.13%	0.8130	84.94%	0.8247	87.02% $(2048, 2048)$	0.8538	88.14% $(7, 7, 7)$	0.8653
OrganCMNIST $28 \times 28, 11,$ 12975/2392/8216	81.28%	0.8007	86.70%	0.8524	75.62% $(4096, 2048, 512, 256)$	0.7398	86.04% $(5, 7, 5)$	0.8472
BloodMNIST $28 \times 28, 8,$ 11959/1712/3421	85.27%	0.8332	90.97%	0.8956	86.76% $(512, 2048, 256)$	0.8518	93.13% $(5, 7, 3)$	0.9249
BloodMNIST $64 \times 64, 8,$ 11959/1712/3421	87.55%	0.8566	93.04%	0.9220	86.03% $(2048, 2048, 256)$	0.8468	94.33% $(3, 5, 7)$	0.9389

Table S11: Accuracies and macro average F1 for AML, AML followed by logistic regression, and the best MLP and XGBoost models selected from cross-validation across tabular datasets. Below the name of each dataset we indicate the train/test split. For the best MLPs, we indicate in parentheses the neuron numbers of the hidden layers of the network.

Dataset	AML		AML & logistic regression		MLP best		XGBoost best	
	accuracy	macro avg. F1	accuracy	macro avg. F1	accuracy	macro avg. F1	accuracy	macro avg. F1
Bank Marketing 38752/6459	89.22%	0.6683	88.05%	0.6938	89.81% (2048, 1024, 512)	0.6736	90.28%	0.7777
Bank Marketing (additional) 35304/5884	91.32%	0.7536	90.58%	0.7552	91.25% (128, 2048, 512, 256)	0.7238	90.74%	0.7878
Otto Product 49502/12376	78.59%	0.7227	79.49%	0.7597	78.37% (2048, 2048, 256)	0.7543	82.50%	0.7991
Amazon Employee Access 29492/3277	95.18%	0.6813	94.87%	0.7365	94.54% (128, 2048, 256)	0.4860	94.45%	0.7380
Seismic bumps 2325/249	91.51%	0.4778	91.51%	0.5194	92.66% (256, 256)	0.4810	90.35%	0.5116
GesturePhaseSegmentation 8885/988	63.26%	0.5860	67.31%	0.6400	60.02% (2048, 2048, 128)	0.5310	72.37%	0.6872
First order theorem proving 5506/612	60.62%	0.5021	58.17%	0.4827	50.16% (2048, 2048, 512, 256)	0.3847	59.64%	0.5047
Philippine 5248/584	70.03%	0.6973	71.92%	0.7177	69.69% (2048, 256, 128)	0.6946	75.51%	0.7551
Phoneme 4863/541	91.13%	0.8879	91.50%	0.8939	87.06% (4096, 2048, 128)	0.8351	91.68%	0.8964
Sylvine 4611/513	93.76%	0.9376	94.54%	0.9454	92.40% (2048, 2048, 128, 256)	0.9240	95.32%	0.9532
Satellite 4590/510	99.41%	0.6985	99.22%	0.7480	99.22% (2048, 2048, 512)	0.6647	99.02%	0.7197
churn 4500/500	90.20%	0.7436	91.20%	0.7917	91.60% (2048, 1024, 128, 256)	0.8190	96.00%	0.9224

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the main claims: subdirect-decomposition-based learning, its atomized-semilattice construction, hidden-rule recovery, finite-data component selection, the Sparse Crossing algorithm, and empirical performance. The decomposition and atomized-semilattice construction are developed in Sections 3.1–3.4; hidden-rule recovery is addressed in Section 3.5; finite-data component selection is analyzed in Section 3.6; Sparse Crossing is introduced in Section 3.7; and the empirical claims are supported by experiments on classification datasets and Hamiltonian-cycle specifications in Sections 4.1 and 4.2.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper includes a separate Limitations section (Section 6). It discusses the limitations in theoretical analysis, task encodings, assumptions in the probabilistic model, runtime limitation for the current implementation, and the limitations in the empirical analysis.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.

- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Each theoretical claim is backed by a corresponding theorem, with assumptions stated in the theorem statement. The main paper states the results needed for the learning framework, and the appendices provide the complete formal statements and proofs.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The empirical results are detailed in a Appendix E 5, 6 and 7. Additionally, for our algebraic method we give pseudocode (Appendix D 13), open source code and notebooks that guide readers through the results (Appendix E 2)

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.

(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: As detailed in Appendix E 2, the paper provides open-source code, notebooks, and instructions for reproducing the main experimental results.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The main paper describes the experimental setup needed to interpret the results. (Appendix E 4-7) provides the full training details, including the fixed settings used for the algebraic method and the hyperparameter-search procedure for the multilayer perceptron baselines. Tables S1 and S2 report the dataset splits and the selected multilayer-perceptron architectures.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The paper reports statistical significance tests for the empirical comparisons. Specifically, it uses paired two-sided Wilcoxon signed-rank tests across the 24 datasets, reports the resulting p-values, and provides Hodges–Lehmann effect-size estimates with 95% confidence intervals for accuracy and macro-averaged F1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix E 1 gives the computer resources used.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms to the NeurIPS Code of Ethics. The experiments use publicly available datasets and do not involve human-subjects research conducted by the authors. The paper also provides the experimental details, pseudocode, open-source code, and notebooks to support transparency and reproducibility.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: The paper is foundational and studies a general algebraic learning principle and algorithm. The experiments are limited to publicly available benchmark datasets and formal problem specifications, and no deployed application or domain-specific decision system is proposed. We therefore do not identify direct societal impacts beyond those associated with general-purpose machine learning methodology.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release high-risk models or datasets with substantial misuse potential. The released assets consist of code and notebooks for reproducing experiments on publicly available benchmark datasets and formal problem specifications.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper cites the original sources for the datasets, software packages, and statistical methods used in the experiments. Appendix E.3 reports the licenses for the existing assets.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The paper introduces new research assets in the form of code and accompanying notebooks. An anonymized URL is provided in Appendix E.2, and the released materials include documentation and instructions for running the code.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: We did not crowdsource and we did not use human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were used only for writing and editing assistance and were not part of the core methodology, experiments, analysis, or scientific claims of the paper.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.