Scalable Query Answering under Uncertainty to Neuroscientific Ontological Knowledge: The NeuroLang Approach

Gaston E Zanitti, Yamil Soto, Valentin Iovene, Maria Vanina Martinez, Ricardo O Rodriguez, Gerardo I Simari, Demian Wassermann

Scalable Query Answering under Uncertainty to Neuroscientific Ontological Knowledge: The NeuroLang Approach

Gaston E. Zanitti[1] , **Yamil Soto**[2] , **Valentin Iovene**[1] , **Maria Vanina Martinez**[3] , **Ricardo O. Rodriguez**[3] , **Gerardo I. Simari**[2], **Demian Wassermann**[1]

[1] INRIA Saclay, Equipe Parietal, 1 Rue Honoré d'Estienne d'Orves, 91120 Palaiseau, France

[2] Dept. of Computer Science and Engineering, Universidad Nacional del Sur (UNS)

[3] Dept. of Computer Science, Universidad de Buenos Aires (UBA)

{gaston.zanitti,valentin.iovene,demian.wassermann}@inria.fr

{yamil.soto,gis}@cs.uns.edu.ar

{mvmartinez,ricardo}@dc.uba.ar

Abstract

Researchers in neuroscience have a growing number of datasets available to study the brain, made possible by recent technological advances. Given the extent to which the brain has been studied, there is also available ontological knowledge encoding the current state of the art regarding its different areas, activation patterns, key words associated with studies, etc. Furthermore, there is an inherent uncertainty associated with brain scans arising from the mapping between voxels—3D pixels—and actual points in different individual brains. Unfortunately, there is currently no unifying framework for accessing such collections of rich heterogeneous data under uncertainty, making it necessary for researchers to rely on ad hoc tools. In particular, one major weakness of current tools that attempt to address this kind of task is that only very limited propositional query languages have been developed. In this paper, we present NeuroLang, an ontology language with existential rules, probabilistic uncertainty, and built-in mechanisms to guarantee tractable query answering over very large datasets. After presenting the language and its general query answering architecture, we discuss real-world use cases showing how NeuroLang can be applied to practical scenarios for which current tools are inadequate.

1 Introduction

Recent technological advances in neuroscience have sparked enormous growth in the amount of datasets—containing text, images, and knowledge graphs—available for analysis of the human brain. To take advantage of the full breadth of this heterogeneous, and often noisy, data, a unifying framework is needed that allows researchers to represent their theories, definitions, and perform inferences on them in a structured, formal way. The main hypothesis of this paper is that a probabilistic Datalog+/– language carefully extended with negation and aggregation is the perfect tool for such task.

One of the central neuroscience use cases requiring the combination of the aforementioned datasets, are meta-analysis tools. This application constitutes a fertile ground to show how current knowledge representation advancements can combine heterogeneous datasets, pushing forward neuroimaging research. Meta-analysis is a set of techniques used to combine a finite number of published articles, which often disagree, to infer consensus-based findings (Poldrack and Yarkoni 2016). Hence, its main application is aggregating noisy knowledge across articles in the field. While recent advances in automated meta-analysis techniques are mostly centered in better representing spatial correlations (Samartsidis et al. 2017), to the best of our knowledge none have formally addressed expressivity limitations of query languages and the feasibility of a more expressive resolution.

Current standard tools in neuroimaging meta-analyses are NeuroSynth and BrainMap (Yarkoni et al. 2011; Laird et al. 2011), which harness automatically-extracted as well as manually-curated information present across neuroscientific articles. Briefly, these tools interpret each article as an independent sample of *neuroscientific knowledge*, and then develop query systems centered on study subset selection and posterior probabilistic inference on such subsets. For instance, selecting all studies mentioning "fear" and inferring the most common areas of the brain reported as active—i.e. deferentially oxygenated—in such studies. In these tools, queries select a subset of a total of around 15k full-text articles reporting involvement of several brain locations each, and a brain tessellation of 300k cubes, or voxels, then infer commonalities across these articles through maximum likelihood estimations combined with spatial information smoothing. Such queries can express questions like "Where do articles reporting the term 'emotion' show activations?", or "Which terms associated with cognitive processes or physiological concepts are most likely associated with articles reporting activations in the amygdala?". Finally, after the inferential tasks, the obtained probabilities are manipulated and aggregated to frame results into the frequentist language neuroscientists commonly use to communicate the significance of their results (Yarkoni et al. 2011; Samartsidis et al. 2017). Through the design of task-specific inferential algorithms, these meta-analyses are performed in under 30 seconds on a regular laptop computer. However, these tools are limited in terms of the expressivity of their associated query languages.

NeuroSynth combines text mining, meta-analysis, and machine learning techniques to generate probabilistic mappings relating text-mined terms with activations in the hu-

man brain. But the language to infer these relationships is based on propositional logic. This limitation excludes, for instance, the use of existentials and negation, forbidding queries such as "What are the terms most probably mentioned in articles reporting activations in the parietal lobe and in no other brain region", which we dub *segregation queries*. Another example of this situation is BrainMap, which has a hand-curated dataset of great precision and an ontology for structuring all this knowledge and annotate the articles, but can only be queried with a very limited propositional logic language that only allows to select terms mentioned in the articles and the leaves of the ontology, which again can't express segregation queries or harness the full information of neuroscience ontologies—such as CogAt (Poldrack et al. 2011)—that use open knowledge.

Breaching the expressivity limitations of current approaches and handling heterogeneous data analysis requires tackling several issues: handling noisiness in neuroimaging data and conclusions reported across studies calls for a unifying formalism with probabilistic modeling capabilities; being able to leverage ontological information that models information under the open world assumption; finally, performance cannot be ignored since the amount of information needed to model the human brain is considerable, and current tools perform inferences in under 30 seconds. For short, we need to design a logic language capable of performing negation and aggregation; performing probabilistic inference; deal with open knowledge; being able to deal with the post-processing of inferred probabilities; and capable of dealing with neuroimaging databases having, at least, a similar performance to current meta-analytic tools.

Our main proposal in this paper is that a subset of Datalog+/–, extended with probabilistic semantics, aggregation, and negation, is a perfect fit for meta-analytic applications. Such an approach allows us to have a language based on first order logic with negation and existentials (FO$^{\neg\exists}$), enabling more complex queries such as segregation queries or manipulation of open-world information. In all, we produce a language that can express the full breadth of the pipeline needed for meta-analytic applications: from data preprocessing to probabilistic modelling and inference, and finally the post-processing of probabilistic results into images and reports that are easily interpretable in terms of current reporting used in neuroscience publications. In this work, we introduce NeuroLang, a probabilistic language based on Datalog+/– developed as a probabilistic domain-specific language for expressing and solving rich logic-based queries meeting the functional requirements of neuroimaging meta-analyses.

The rest of this paper is organized as follows: Section 2 introduces the probabilistic semantics, which is based on a classical possible world approach adopted in many approaches to reasoning under uncertainy; Section 3 then formally introduces the NeuroLang language and the NEUROLANGQA query answering algorithm; Section 4 presents a set of real-world use cases showing how our formalism can be applied in neuroscientific research; finally, Section 5 discusses conclusions.

2 Basic Probabilistic Ontological Model

In this section, we recall the basics on relational databases, conjunctive queries, Datalog, and ontology-mediated query answering (including tuple-generating dependencies and negative constraints), all based on a basic probabilistic extension with a corresponding query answering semantics.

We assume an infinite universe of *(data) constants* Δ, an infinite set of *(labeled) nulls* Δ_N (used as "fresh" Skolem terms) that are placeholders for unknown values, and an infinite set of variables $\mathcal{V}$. Different constants represent different values (*i.e., unique name assumption*), while different nulls may represent the same value. Sequences of $k \geq 0$ variables, namely $X_1, \ldots, X_k$, are denoted by $\mathbf{X}$.

Furthermore, we assume a *relational schema* $\mathcal{R}$, which is a finite set of *predicate symbols*, we also allow built-in predicates (with finite extensions) and equality. As expected, a *term* t is a constant, null, or variable. An *atomic formula* (or *atom*) $\mathbf{a}$ has the form $p(t_1, \ldots, t_n)$, where p is an n-ary predicate, and $t_1, \ldots, t_n$ are terms. We denote with $\mathcal{F}$ the set of all ground atoms built from $\mathcal{R}$ and Δ. A negated atom is of the form $\neg a$ where a is an atom. We are going to assume that $\mathcal{R} = \mathcal{R}_D \cup \mathcal{R}_P$, with $\mathcal{R}_D \cap \mathcal{R}_P = \emptyset$, containing predicates that refer to deterministic and probabilistic events, respectively.

A *database instance* D for a relational schema $\mathcal{R}_D$ is a (possibly infinite) set of atoms with predicates from $\mathcal{R}_D$ and arguments from Δ. On the other hand, let a *probabilistic atom* be of the form $\mathbf{a} : p$, where p is a real number in the interval $[0, 1]$ and $\mathbf{a}$ is an atom with a predicate from $\mathcal{R}_P$. We do not allow negation in probabilistic atoms.

A *probabilistic constraint* c has the form

$$\mathbf{a}_1 : p_1 \mid \ldots \mid \mathbf{a}_k : p_k,$$

where $k > 0$, each $\mathbf{a}_i : p_i$ is a probabilistic atom, and $\sum p_i \leq 1$. If the p_i's in a probabilistic constraint do not sum to 1, then there exists also the possibility that none of them happen. The probability of this complementary event is $1 - \sum p_i$. Given a probabilistic constraint $c = \mathbf{a}_1 : p_1 \mid \ldots \mid \mathbf{a}_k : p_k$, we will make use of the notation $\mathit{atoms}(c) = \{\mathbf{a}_1, \ldots, \mathbf{a}_k\}$. We will also denote the probability of any atom $\mathbf{a}$ with $p(\mathbf{a})$. We have that $p(\mathbf{a}_i) = p_i$ whenever $\mathbf{a}_i : p_i$ belongs to a probabilistic constraint c.

Given a set of probabilistic constraints C, note that each ground atom can only appear in one constraint in C. This approach is similar to *probabilistic databases* (see (Suciu et al. 2011)) where each tuple comes from a general probability distribution over tuples and inexistence is one of the options. This allows to incorporate beliefs about the likelihood of tuples and cell values. From a practical point of view, we will see that this assumption restricts the number of possible worlds by limiting the potential combinations. In (Vennekens, Denecker, and Bruynooghe 2009, Eq. 5), the proposed semantics is more complex and this assumption is relaxed.

Example 1. *Consider the following example where we have a database instance D and a set of probabilistic constraints C (recall that t_i atoms cannot appear in C).*

$$D = \{t_1(a), t_1(c), t_2(a), t_2(b)\}$$

$$C = \left\{ \begin{array}{llll} c_1 = s(a,b) & : 0.3 & & \\ c_2 = s(b,c) & : 0.7 & & \\ c_3 = r(b) & : 0.4 & | \quad r(c) & : 0.1 \end{array} \right\}$$

Tuple Generating Dependencies Given a relational schema $\mathcal{R}$, a *tuple-generating dependency (TGD)* σ is a first-order formula of the form:

$$\forall \mathbf{X} \forall \mathbf{Y}\, \Phi(\mathbf{X}, \mathbf{Y}) \rightarrow \exists \mathbf{Z}\, \Psi(\mathbf{X}, \mathbf{Z}),$$

where $\Phi(\mathbf{X}, \mathbf{Y})$ and $\Psi(\mathbf{X}, \mathbf{Z})$ are conjunctions of atoms over $\mathcal{R}$ (without nulls), called the *body* and the *head* of σ, denoted $body(\sigma)$ and $head(\sigma)$, respectively. Such σ is satisfied in a database D for $\mathcal{R}$ if and only if, whenever there exists a homomorphism h that maps the atoms of $\Phi(\mathbf{X}, \mathbf{Y})$ to atoms of D, there exists an extension h' of h that maps the atoms of $\Psi(\mathbf{X}, \mathbf{Z})$ to atoms of D. All sets of TGDs are finite here and we assume without loss of generality that every TGD has a single atom in its head. Furthermore, we say that a TGD σ is *full* whenever there are no existential variables in the head.

Example 2. *Based on Example 1 we can add the following set of rules:*

$$\begin{aligned} \Sigma = \{&\forall X\ t_1(X) \rightarrow \exists Z\ o(X, Z), \\ &\forall X \forall Y\ t_2(X) \wedge o(X, Y) \rightarrow t(X), \\ &\forall X \forall Y\ s(X, Y) \wedge r(Y) \rightarrow w(X, Y)\} \\ A = \{&\forall X \forall W\ v(X, W) \rightarrow u(X, \max(W))\} \end{aligned}$$

TGDs can be extended to allow negation—in this work we only allow stratified negation (Abiteboul, Hull, and Vianu 1995) for full TGDs. Furthermore, as shown by the rule in set A in the previous example, we extend the language so aggregation functions can be used in the head of full TGDs (Abiteboul, Hull, and Vianu 1995). As we see in the following section, we restrict the syntax of this type of rules so that neither negation nor recursion is allowed.

Definition 1. *A* probabilistic ontology $\mathcal{O} = (D, C, \Sigma)$ *consists of a database instance D, a set C of probabilistic constraints, and a set Σ of arbitrary TGDs.*

Note that a database instance can be thought of as a set of probabilistic constraints with only probabilistic atoms, each one annotated with probability 1. Furthermore, the structure (D, Σ) corresponds to a knowledge base with existential rules as defined in (Calì, Gottlob, and Lukasiewicz 2012), whenever rules in Σ do not involve atoms that appear in probabilistic constraints. We will see in the following that the probabilistic semantics for query answering presented in this section naturally extends the classical semantics in (Calì, Gottlob, and Lukasiewicz 2012).

Semantics. We take the notion of possible world (or interpretation) of a probabilistic ontology as a subset of $\mathcal{F}$ and we denote with Ω the set of all possible worlds. Each possible world $\omega \in \Omega$ satisfies the following property:

$$\forall F \in \mathcal{F} : \omega \models F \text{ iff } F \in \omega; \qquad \text{otherwise } \omega \models \neg F$$

This means that ω is a complete interpretation of every element of $\mathcal{F}$. The usual semantics of a classical Datalog program P is the least Herbrand model that contains exactly all ground facts in P plus every ground atom inferred from it, i.e. the intersection of all worlds that satisfy P.

However, in the probabilistic case we need to consider a generalization of this semantics so that every ground fact has associated a probability value. According to this idea, we are going to take the models of a set of non-probabilistic ontologies, induced by total choices, so that they all share the same TGDs but the corresponding database instances differ. As mentioned before, in our approach we have two ways of associating probability to facts. In the first one, a fact corresponds to a Boolean random variable that is true with probability p and false with probability $1 - p$. In the second, we interpret facts as multi-valued random variables instead of binary ones. We use probabilistic constraints for representing both, and assume that the facts within the same constraint are mutually exclusive events, where facts in different constraints are mutually independent events. According to this idea, we give the following definition:

Definition 2. *Given a probabilistic ontology $\mathcal{O} = (D, C, \Sigma)$, for each $1 \leq j \leq |C| : c^j = \mathbf{a}_1^j : p_1^j \mid \ldots \mid \mathbf{a}_k^j : p_k^j$, with $c^j \in C$, we have:*

$$\textit{choices}(c^j) = \{\mathbf{a}_i^j \mid 1 \leq i \leq k\} \cup \{\bot_{c^j}\}.$$

For each $b = \mathbf{a}_i^j \in c^j$, we have $p(b) = p_i^j$ and $p(\bot_{c^j}) = 1 - \sum_{1 \leq i \leq k} p_i^j$.

The set of total choices for $\mathcal{O}$ is defined as total_choices$(C) =$

$$\{[b_1, \ldots, b_l] \mid l = |C|, 1 \leq j \leq |C| : b_j \in \textit{choices}(c^j)\}.$$

The probability of a particular total choice $\lambda \in$ total_choices(C) is defined as $p(\lambda) = \prod_{1 \leq j \leq l}^{[b_1, \ldots, b_l] \in \lambda} p(b_j)$. We use notation atoms$(\lambda) = \{\mathbf{b}_j \neq \bot_{c^j} \mid 1 \leq j \leq l : [b_1, ..., b_l] \in \lambda\}$ and atoms$(C) = \bigcup_{\lambda \in \textit{total_choices}(C)}$ atoms(λ).

Definition 3. *Let ω and λ be a possible world and a total choice, respectively. Then, we will say that ω satisfies λ, denoted $\omega \models \lambda$, if and only if atoms$(\lambda) \subseteq \omega$. Also, $\|\lambda\|$ will denote the set of possible worlds of a total choice, i.e. $\|\lambda\| = \{\omega \in \Omega \mid \omega \models \lambda\}$.*

Example 3. *The set of all total choices for probabilistic ontology (D, C, Σ) from Examples 1 and 2 is the following:*

$$\begin{array}{llllll} \lambda_1 & = [s(a,b), & s(b,c), & r(b)] & p(\lambda_1) & = 0.084 \\ \lambda_2 & = [s(a,b), & \bot_{c_2}, & r(b)] & p(\lambda_2) & = 0.036 \\ \lambda_3 & = [\bot_{c_1}, & s(b,c), & r(b)] & p(\lambda_3) & = 0.196 \\ \lambda_4 & = [\bot_{c_1}, & \bot_{c_2}, & r(b)] & p(\lambda_4) & = 0.084 \\ \lambda_5 & = [s(a,b), & s(b,c), & r(c)] & p(\lambda_5) & = 0.021 \\ \lambda_6 & = [s(a,b), & \bot_{c_2}, & r(c)] & p(\lambda_6) & = 0.009 \\ \lambda_7 & = [\bot_{c_1}, & s(b,c), & r(c)] & p(\lambda_7) & = 0.049 \\ \lambda_8 & = [\bot_{c_1}, & \bot_{c_2}, & r(c)] & p(\lambda_8) & = 0.021 \\ \lambda_9 & = [s(a,b), & s(b,c), & \bot_{c_3}] & p(\lambda_9) & = 0.105 \\ \lambda_{10} & = [s(a,b), & \bot_{c_2}, & \bot_{c_3}] & p(\lambda_{10}) & = 0.045 \\ \lambda_{11} & = [\bot_{c_1}, & s(b,c), & \bot_{c_3}] & p(\lambda_{11}) & = 0.245 \\ \lambda_{12} & = [\bot_{c_1}, & \bot_{c_2}, & \bot_{c_3}] & p(\lambda_{12}) & = 0.105 \end{array}$$

It is easy to see that *total_choices*(C) defines a partition on Ω by using the following equivalence relation on $\Omega \times \Omega$: $\omega \equiv \omega'$ if and only if $\forall \lambda \in$ *total_choices*$(C) : \omega \models \lambda \Leftrightarrow \omega' \models \lambda$.

We define the semantics of a probabilistic ontology based on the semantics of a classical ontology with existential rules (TGDs). Intuitively, each total choice induces a classical (i.e., non-probabilistic) ontology.

Definition 4. *Let $\mathcal{O} = (D, C, \Sigma)$, be a probabilistic ontology, and let λ be a total choice of C. Then, the (non-probabilistic) ontology induced by $\lambda = [b_1, \dots b_l]$ is defined as $O_\lambda = (D_\lambda, \Sigma)$, with $D_\lambda = D \cup \{b_1, \dots b_l\}$.*

Example 4. *Based on the total choices from Example 3 and probabilistic ontology $\mathcal{O} = (D, C, \Sigma,)$, each λ_i with $1 \leq i \leq 12$, induces a non-probabilistic ontology $\mathcal{O}_{\lambda_)} = (D_{\lambda_i}, \Sigma)$ where $D_{\lambda_i} = \mathcal{D} \cup \{b_1, \dots, b_l\}$ with $b_k \in \lambda_i$ and $b_k \neq \bot_{c_j}$ for every $c_j \in C$.*

We recall the notion of models and satisfaction for classical ontologies (Calì, Gottlob, and Lukasiewicz 2012).

Definition 5. *Given an ontology (D, Σ), the set of* models*, denoted $mods(D, \Sigma)$, is the set of all (possibly infinite) databases B such that (i) $D \subset B$, and (ii) every $\sigma \in \Sigma$ is satisfied in B.*

Note that each B in the above definition can be considered as a possible world under the closed world assumption, i.e. every tuple that does not appear in B is false. It is important to recall that for full TGDs (pure Datalog rules), an ontology (D, Σ) has a unique least model (Abiteboul, Hull, and Vianu 1995).

Definition 6. *Let $\mathcal{O}$ be a probabilistic ontology, and Φ be a conjunction of ground atoms built from predicates in $\mathcal{R}$. The probability that Φ holds in $\mathcal{O}$, denoted $Pr^{\mathcal{O}}(\Phi)$, is the sum of the probabilities of all total choices λ such that $(D_\lambda, \Sigma) \models \Phi$; that is, $Pr^{\mathcal{O}}(\Phi) = \sum_{(D_\lambda, \Sigma) \models \Phi}^{\lambda \in \text{total\_choice}(C)} p(\lambda)$.*

At this point, it is interesting to remark the connection between our approach and the one considered by Rigguzzi et al. in (Riguzzi 2008; Riguzzi 2006). The Logic Programs with Annotated Disjunctions (LPADs) mentioned in their paper make an implicit treatment of mutually exclusive facts, whereas our approach does it explicitly. In fact, LPADs are more expressive than our language since they use non-Horn clauses. In addition, they use well-founded semantics in order to deal with negation as failure. Both aspects have a computational cost that we wish to avoid.

Semantics for Query Answering. A *conjunctive query (CQ)* over $\mathcal{R}$ has the form $Q(\mathbf{X}) = \exists \Phi(\mathbf{X}, \mathbf{Y})$, where $\Phi(\mathbf{X}, \mathbf{Y})$ is a conjunction of atoms (possibly equalities, but not inequalities) with the variables $\mathbf{X}$ and $\mathbf{Y}$, and possibly constants, but without nulls. Probabilistic answers to CQs are defined via *homomorphisms*, which are mappings $\mu \colon \Delta \cup \Delta_N \cup \mathcal{V} \to \Delta \cup \Delta_N \cup \mathcal{V}$ such that (i) $c \in \Delta$ implies $\mu(c) = c$, (ii) $c \in \Delta_N$ implies $\mu(c) \in \Delta \cup \Delta_N$, and (iii) μ is naturally extended to atoms, sets of atoms, and conjunctions of atoms.

Definition 7. *The set of all* probabilistic answers *to a CQ $Q(\mathbf{X}) = \exists \mathbf{Y}\, \Phi(\mathbf{X}, \mathbf{Y})$ over a probabilistic ontology $\mathcal{O} = (D, C, \Sigma)$, denoted with $ans(Q, D, C, \Sigma)$, or $ans(Q, \mathcal{O})$, is a set of pairs (t, p_t) with t a tuple over Δ such that there exists a homomorphism $\mu \colon \mathbf{X} \cup \mathbf{Y} \to \Delta \cup \Delta_N$ with $\mu(X) = t$ and $(D_\lambda, \Sigma) \models \mu(\Phi(\mathbf{X}, \mathbf{Y}))$ for all $\lambda \in$ total_choice(C). The probability of each tuple t is then $p_t = \sum_{(D_\lambda, \Sigma) \models \mu(\Phi(\mathbf{X}, \mathbf{Y}))}^{\lambda \in \text{total\_choice}(C)} p(\lambda)$.*

Observation. If a probabilistic ontology $\mathcal{O} = (D, C, \Sigma)$ is such that C is empty, then the semantics for (B)CQs as defined above coincides with that for classical ontologies (Calì, Gottlob, and Lukasiewicz 2012).

Note that query answering under general TGDs for non-probabilistic ontologies is undecidable (Beeri and Vardi 1981), even when the schema and TGDs are fixed (Calì, Gottlob, and Kifer 2008). The two problems of CQ and BCQ evaluation under TGDs are LOGSPACE-equivalent (Fagin et al. 2005a; Deutsch, Nash, and Remmel 2008). As mentioned above, in the non-probabilistic case, for arbitrary full TGDs there exists exactly one minimal model (Abiteboul, Hull, and Vianu 1995) over which Q is evaluated. Furthermore, it has been shown that for full TGDs CQ evaluation can be done in polynomial time in data complexity (*i.e., assuming σ and Q fixed*) (Dantsin et al. 2001).

3 NeuroLang Programs

We assume the existence of a separate schema $\mathcal{T}$, the target schema, that defines the language by means of which users of NeuroLang can query about the probability of certain events. Predicates in $\mathcal{T}$ have a distinguished term in the n-th position (for n-ary predicates) reserved exclusively for real numbers in the interval $[0, 1]$; i.e., for any predicate $p \in \mathcal{T}$, atoms of the form $p(a_1, \dots, a_n)$ are such that $a_1, \dots, a_{n-1}$ are variables or constants from Δ, while a_n is a variable or a constant from $[0, 1]$. Below we show an example of how this language is used.

A NeuroLang program $\mathcal{N}$ is comprised of the following components:

- D, Σ: where D is a set of ground atoms from $\mathcal{R}_D$, and Σ is a set of full TGDs that only use atoms from $\mathcal{R}_D$ and can have recursion and stratified negation.
- (D_1, Σ_1): a classical ontology, where D_1 is a set of ground atoms from $\mathcal{R}_D$, Σ_1 is a set of TGDs that belong to the Sticky fragment, and the bodies and heads are atoms built from predicates in $\mathcal{R}_D$.
- C: a set of probabilistic constraints only involving atoms from $\mathcal{R}_P$.
- χ: a set of full TGDs, whose bodies and heads may contain atoms from $\mathcal{R}_D \cup \mathcal{R}_P$. Neither negation nor recursion is allowed in this set of rules.
- Π: a set of *probability encoding rules* (PERs) with the following form:

$$\sigma^* : \forall \mathbf{X} \forall \mathbf{Y} (\Phi(\mathbf{X}, \mathbf{Y})) \to \psi(\mathbf{X}, \rho_X)$$

where Φ is a conjunction of atoms from $\mathcal{R}_D \cup \mathcal{R}_P$, ψ is a atom in $\mathcal{T}$ and ρ_X is the distinguished term that in this

case can only be a variable (ranging over the real interval $[0, 1]$).

- A: a set of rules of the form

$$\forall \mathbf{X} \forall \mathbf{Y} (\Phi(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) \rightarrow \psi(\mathbf{X}, agg(\mathbf{Z})) \quad (1)$$

where Φ is a conjunction of atoms in $\mathcal{R}_D \cup \mathcal{T}$ and *agg* is an aggregation function (e.g., sum, count, avg, etc.). Neither negation nor recursion is allowed in this set of rules.

Intuitively, the above sets together provide the following functionalities:

(i) Σ, Σ_1, $\mathcal{C}$, and χ are used by the probabilistic inference mechanism, which applies ontological rules and ultimately associates probabilities to atoms (following the semantics described in Section 2);

(ii) Φ incorporates probabilities as values inside atoms; and

(iii) rules in A manipulate these probabilities via aggregation functions in order to present them as requested by the user.

Note that PERs are full TGDs, but they will be used *outside of the logic* to translate from a source schema to a target one, in the same spirit as source-to-target TGDs for data exchange (Fagin et al. 2005b). On the other hand, for rules in A we incorporate functional symbols *agg* to the distinguished term in ψ to indicate that its value takes the result of applying the function *agg* to all ρ_X that satisfy the body of the rule. Note that users here can define arbitrary rules that manipulate probabilities by means of aggregation functions. As in the case of PERs, this is also outside of the logic since it is defined as a post-processing step that builds a view as defined by the user issuing the query. We extend notation *body* and *head* used for TGDs to all types of rules defined in this section.

The following is a simple example of query answering using PERs.

Example 5. *From the previous examples we can build the following NeuroLang program $\mathcal{N}$. We add a set of PERs and rules with aggregations.*

$$\begin{aligned}
D_1 &= \{t_1(a), t_1(c)\},\\
\Sigma_1 &= \{\forall X t_1(X) \rightarrow \exists Z\ o(X, Z)\},\\
D &= \{t_2(a), t_2(b)\},\\
\Sigma &= \{\forall X \forall Y\ t_2(X) \wedge o(X, Y) \rightarrow t(X)\},\\
C &= \left\{ \begin{array}{llll} s(a,b) & : 0.3 & & \\ s(b,c) & : 0.7 & & \\ r(b) & : 0.4 & | \quad r(c) & : 0.1 \end{array} \right\},\\
\chi &= \{\forall X \forall Y\ s(X, Y) \wedge r(Y) \rightarrow w(X, Y)\}\},\\
\Pi &= \{\forall X \forall Y\ w(X, Y) \rightarrow v(X, \rho_X)\},\\
A &= \{\forall X \forall W\ v(X, W) \rightarrow u(\max(W))\},\\
Q_1(X, P) &= v(X, P), t(X),\\
Q_2(X, P) &= v(X, P), u(P).
\end{aligned}$$

Therefore, the partition of possible worlds used to compute queries Q_1 and Q_2—excluding atoms coming from D and (D_1, Σ_1) for clarity, and including probabilities—is the following:

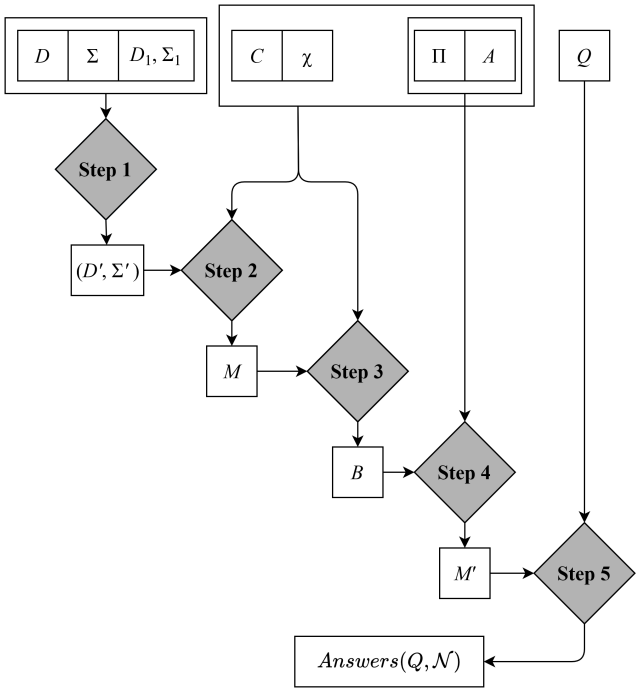

Figure 1: Overview of the NEUROLANGQA algorithm. Step numbers refer to those described in Algorithm 1

$$\left\{ \begin{array}{llllll}
\{s(a,b) & s(b,c) & w(a,b) & r(b) & t(a)\} & : 0.084\\
\{s(a,b) & & w(a,b) & r(b) & t(a)\} & : 0.036\\
\{ & s(b,c) & & r(b) & t(a)\} & : 0.196\\
\{ & & & r(b) & t(a)\} & : 0.084\\
\{s(a,b) & s(b,c) & w(b,c) & r(c) & t(a)\} & : 0.021\\
\{s(a,b) & & & r(c) & t(a)\} & : 0.009\\
\{ & s(b,c) & w(b,c) & r(c) & t(a)\} & : 0.049\\
\{ & & & r(c) & t(a)\} & : 0.021\\
\{s(a,b) & s(b,c) & & & t(a)\} & : 0.105\\
\{s(a,b) & & & & t(a)\} & : 0.045\\
\{ & s(b,c) & & & t(a)\} & : 0.245\\
\{ & & & & t(a)\} & : 0.105
\end{array} \right\}$$

Answering Q_1, Q_2 leads to the target schema solution $\{v(a, 0.141), v(b, 0.154), u(0.154)\}$. Hence, the resulting answer set is $\{Q_1(a, 0.141), Q_2(b, 0.154)\}$.

Query Answering in NeuroLang

A *NeuroLang query* Q is any conjunction of atoms in $\mathcal{R}_D \cup \mathcal{T}$, such that atoms in $\mathcal{T}$ have as distinguished term a variable; these variables will be instantiated with the probability of certain events as computed by the inference mechanism.

Algorithm 1 describes the pseudocode for answering queries in the NeuroLang framework—Figure 1 provides a high-level view of the main steps involved in this process, where inputs are as defined above.

There are two steps in which NEUROLANGQA makes external calls. First, in Step 1 the rewriting of Σ w.r.t.

Algorithm 1: NEUROLANGQA

Input : NeuroLang program $\mathcal{N} = (D, \Sigma, (C, \chi), (D_1, \Sigma_1), \Pi, A)$ and query $Q(\mathbf{X}) = \exists \mathbf{Y} \Phi(\mathbf{X}, \mathbf{Y})$
Output: $ans(Q(\mathbf{X}), \mathcal{N})$

1 **Step 1:** Obtain database instance D' and set of full TGDs Σ' such that $D^{'} = D \cup D_1$ and $\Sigma^{'}$ is the rewriting of Σ with respect to Σ_1.
2 **Step 2:**
3 2a: Let *Aux* be the set of TGDs in $\Sigma^{'}$ whose bodies do not depend on $C \cup \chi \cup \Pi$.
4 2b: Let M the set of ground atoms a s.t. $(D', \mathit{Aux} \cup A) \models a$
5 **Step 3:**
6 B:= $\emptyset$
7 **foreach** *PER* $\pi \in \Pi$ **do**
8 Let $Q_\pi(\mathbf{X}) = \mathit{body}(\pi)$ `// Rule bodies are taken as queries`
9 *probAnsPairs*:= $ans(Q_\pi(\mathbf{X}), (M, C, \chi))$ `// Obtain probability values associated with each query answer`
10 **foreach** $(t, p) \in$ *probAnsPairs* **do**
11 Let h' be the instantiation of $\mathit{head}(\pi)$ with values from (t, p)
12 B:= $B \cup \{h', Q_\pi(t)\}$ `// Add query answers and PER heads to set` B
13 **end**
14 **end**
15 **Step 4:** Let M' the set of ground atoms a such that $(B, (\Sigma' - Aux) \cup A) \models a$
16 **Step 5:** Return $ans(Q(\mathbf{X}), \mathcal{N})$ computed from atoms in M'.

Σ_1 is done by means of the XRewrite algorithm developed in (Gottlob, Orsi, and Pieris 2014) for rewriting queries with respect to the Sticky fragment of existential rules (also known as Datalog+/–). Note that here, the algorithm is used to rewrite every appearance of heads of rules in Σ_1 in the bodies of rules in Σ, yielding a potentially larger set of full TGDs (rules without existentials in the head).

Then, Step 3 derives the probabilities associated with atoms (Line 9). This is done by dynamically choosing the best algorithm for the job; if π is liftable according to (Dalvi and Suciu 2012), then lifted query answering is applied; otherwise, the query is compiled to an SDD representation and model counting is applied (Vlasselaer et al. 2014). Both cases are implemented in relational algebra with provenance (Senellart 2017).

The final step of the algorithm returns the answers to query Q as the set of all tuples t built from Δ such that there exists a homomorphism μ where $\mu(\mathbf{X}) = t$ and $\mu(\Phi(\mathbf{X}, \mathbf{Y})) \in M'$.

Correctness of NEUROLANGQA**.** We now discuss the correctness of algorithm NEUROLANGQA with respect to the probabilistic semantics described in Section 2.

Without loss of generality, we assume a query of the form $Q(\mathbf{X}, \rho_\mathbf{X}) = \Phi(\mathbf{X}) \wedge \psi_i(\mathbf{X}, \rho_\mathbf{X})$, where $\Phi(\mathbf{X})$ is a conjunction of atoms in $\mathcal{R}_D$ and $\psi_i(\mathbf{X}, \rho_\mathbf{X})$ is an atom in $\mathcal{T}$.

The result of Step 1 in NEUROLANGQA is a special case of a probabilistic ontology (D', Σ'), where Σ' is a set of full TGDs that may contain stratified negation and recursion. Furthermore, Step 2a removes from Σ' all rules that depend (Baget et al. 2011) on $C \cup \chi \cup \Phi$. Therefore, M computed in Step 2b is unique as neither probabilistic atoms, nor existential rules are involved. Step 3 now considers the probabilistic ontology defined by $\mathcal{O} = (M, C \cup C', \chi)$. Note that atoms in M materialize ontology (D', Aux) and they will hold in every possible world for probabilistic ontology $\mathcal{O}$.

Recall that the purpose of PERs is to incorporate the probability of an atom as an additional term—Step 3 does precisely that: for each PER π, it computes the probability of all ground instantiations of $\mathit{body}(\pi)$ that are entailed by $\mathcal{O}$. For each such instantiation t, set B contains the instantiation itself ($Q_\pi(t)$) and the head of π instantiated by values in t and an extra position with value $Pr^{\mathcal{O}}(\mathit{body}(\pi)(t))$.

Finally, Step 4 considers a deterministic ontology comprised by B (a set of ground atoms) and the set of full TGDs $(\Sigma' - Aux) \cup A$; M' contains all ground atoms that are entailed by such ontology. As in the case of M, M' is unique since neither existential rules nor probabilistic atoms are involved.

Therefore, we can conclude that—by construction—the results computed by the NEUROLANGQA algorithm are correct with respect to the probabilistic semantics defined in Section 2 up to Step 3. This means that the probabilities associated with atoms in B correspond to the probability with which they are entailed by the probabilistic ontology. The final two steps simply follow the user-specified rules for establishing personalized views, which may manipulate probability values in an arbitrary fashion.

With the framework in place, in the following section we show how it can be applied in practice.

4 Evaluation based on Real-World Use Cases in Neuroscience Research

In this section, we illustrate via concrete examples several use cases that appear in real-world tasks carried out by neuroscience researchers. Since all of our analyses are based on meta-analytic components, we first give a brief description of the NeuroSynth database we use in our examples. Where

extra data is used, it will be clarified in each particular case.

The NeuroSynth database is composed of $3,228$ terms, $14,370$ studies (*SelectedStudy*), and $33,593$ voxels; but this information would not be useful without associations, so we also have $1,049,299$ terms reported as present in studies (*TermInStudy*) and $507,891$ voxels reported as active (*FocusReported*), also with their respective study. Finally there are 112 brain regions from Destrieux's atlas (Destrieux et al. 2010) associated to brain coordinates through the *VoxelByRegionDestrieux* relation. These data give rise to the following extensional databases:

$$D_1 = \left\{ \begin{array}{l} \mathit{TermInStudy}(\text{“}\mathit{emotion}\text{”}, s_1), \\ \mathit{TermInStudy}(\text{“}\mathit{pain}\text{”}, s_1), \\ \vdots \\ \mathit{TermInStudy}(\text{“}\mathit{emotion}\text{”}, s_{4,271}), \\ \vdots \\ \mathit{TermInStudy}(..., s_{14,370}), \\ \\ \mathit{FocusReported}(5, -5, 3, s_1), \\ \mathit{FocusReported}(-10, 5, 1, s_1), \\ \vdots \\ \\ \mathit{VoxelByRegionDestrieux}(15, 47, 16, \\ \quad \mathit{'l\_g\_and\_s\_frontomargin'} \\ ), \\ \mathit{VoxelByRegionDestrieux}(16, 46, 15, \\ \quad \mathit{'l\_g\_and\_s\_frontomargin'} \\ ), \\ \vdots \end{array} \right\},$$

$$C = \left\{ \begin{array}{l} \mathit{SelectedStudy}(s_1) : \frac{1}{14,370} | \ldots \\ \ldots | \mathit{SelectedStudy}(s_{14,370}) : \frac{1}{14,370} \\ \\ \mathit{FocusCoactivates}(\ 5, -5,\ 3, \quad\ 5, -5,\ 3) : 1 \\ \mathit{FocusCoactivates}(\ 5, -5,\ 3, \quad -10,\ \ 5,\ 1) : \\ \qquad (2\pi 2)^{-3/2} \exp\left(-\frac{1}{2} \frac{\|(5,-5,3)-(-10,5,1)\|^2}{2^2}\right) \\ \qquad \vdots \end{array} \right\},$$

where *FocusCoactivates* represents spatial uncertainty in foci reporting, as they encode that the probability that two foci co-activate is mediated by their distance as measured by a 3D Gaussian law with standard deviation of 2mm. This dataset has approximately 5 million atoms as foci further than 8mm away have a co-activation probability of 0. Furthermore, the CogAt ontology (Poldrack et al. 2011) is composed of 56,807 rules.

In the following, the examples are noted in extended Datalog syntax, as in our implemented tool[1].

4.1 Forward inference

In this task, we want to assess the probability of a voxel being reported as active in a study given that word "emotion" is present in the specific studyNote that in order to represent this knowledge we only need the expressive power of full TGDs (no existential rules are needed).

```
% Rules depending on probabilistic constraints
TermAssociation(t) :- SelectedStudy(s), TermInStudy(t, s).

Activation(i, j, k) :-
    SelectedStudy(s),
    FocusReported(i1, j1, k1, s),
    FocusCoactivates(i, j, k,   i1, j1, k1).

% Probability Encoding Rule, where PROB is used to
% encode probability as defined in Section~3, and
% the | operator is syntactic sugar for computing
% conditional probability
% as P(A|B) = P(A,B) / P(B)
ProbMap(i, j, k, PROB) :-
    Activation(i, j, k)
    | TermAssociation("emotion").

% Aggregation to build a single 3D image with the
% probability p as value in each position i, j, k
% keeping only positions with the top 95% of probability
Percentile_95(compute_percentile(p, 95)) :-
    ProbMap(i, j, k, p)

ProbabilityImage(create_region_overlay(i, j, k, p)) :-
    ProbMap(i, j, k, p), Percentile_95(p95), p > p95

% Query that will produce a single image with a
% probability for each voxel
Ans(x) :- ProbabilityImage(x)
```

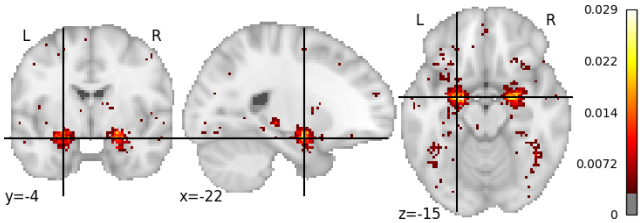

Figure 2: Resulting thresholded brain image from the NeuroLang use case showing that foci in the amygdala are most probably reported if a study includes the word "emotion". As expected the main area shown corresponds to the amygdala (Mesulam 1998).

In Figure 2 we can see how the most important reported activations are concentrated in the amygdala, the region most related to emotions, as generally-accepted in the neuroscience field.

4.2 Reverse inference over a region of the Destrieux atlas

In this task, we will use reverse inference techniques to obtain the terms most likely to be associated with the *short insular gyrus* of the Destrieux atlas. *Atlases* are parcellations of the brain into distinct areas based on histological, physiological, or other characteristics. In this case, the Destrieux atlas separates the most important regions by taking as separation lines the most relevant sulci and gyri.

[1] https://neurolang.github.io

Unfiltered results

Term	Prob
task	0.47
magnetic	0.47
resonance	0.47
magnetic resonance	0.47
functional magnetic	0.43
using	0.38
frontal	0.37
anterior	0.35
network	0.34
prefrontal	0.33

Filtered results

Term	Prob
memory	0.20
attention	0.14
working memory	0.09
perception	0.09
learning	0.08
language	0.08
emotion	0.07

Table 1: Results from experiments section 4.2 and section 4.3. Left: Results (10 of 161 most relevant terms in the top 0.5% most probable terms) of applying reverse inference on region *short insular gyri* of the Destrieux atlas using NeuroSynth term association. Results includes common terms of no importance in terms of cognitive tasks such as "magnetic resonance". Right: Results (0.5% most probable terms) of the application of reverse inference in region *short insular gyri* of the Destrieux atlas using NeuroSynth term association and filtering using the terms present in the CogAt ontology which are in deeper agreement with cognitive function of the short insular gyri (Nieuwenhuys 2012)

```
RegionActivated(s) :-
    VoxelByRegionDestrieux(i, j, k, "l_g_insular_short"),
    FocusReported(i1, j1, k1, s),
    FocusCoactivates(i,j,k,   i1,j1,k1).

RegionActivated(s) :-
    VoxelByRegionDestrieux(i, j, k, "r_g_insular_short"),
    FocusReported(i1, j1, k1, s),
    FocusCoactivates(i,j,k,   i1,j1,k1).

TermProbability(t, PROB) :- TermAssociation(t)
    | RegionActivated(s), SelectedStudy(s).

Percentile_95(compute_percentile(p, 95)) :-
    TermProbability(t, p).

Ans(t, p) :-
    TermProbability(t, p), percentile_95(p95), p > p95.
```

As a result of this example, we can observe in table 1, on the left, the most important terms related to the selected region. It is important to note that the relevant terms in this case are not very useful, because most of them are terms common to all brain studies. In the following use case, we will show how we can improve these results through the use of ontologies. Solving this query takes approximately 4.7 seconds

4.3 Reverse inference over a region of the Destrieux atlas leveraging the CogAt ontology

For this use case, we will again extend the previous one and use the information stored in the CogAt ontology to filter the terms from the reverse inference in order to obtain cleaner results. Terms included in the CogAt ontology are characterized by the "label" relation. The CogAt ontology rewriting adds 4,577 formulas to our database.

```
FilteredTerms(s, t) :- TermInStudy(s, t), label(uri, t).

RegionActivated(s) :-
    VoxelByRegionDestrieux(i, j, k, "l_g_insular_short"),
    FocusReported(i1, j1, k1, s),
    FocusCoactivates(i, j, k,   i1, j1, k1).

RegionActivated(s) :-
    VoxelByRegionDestrieux(i, j, k, "r_g_insular_short"),
    FocusReported(i1, j1, k1, s),
    FocusCoactivates(i, j, k,   i1, j1, k1).

TermProbability(t, PROB) :-
    FilteredTerms(s, t),
    | RegionActivated(s)), SelectedStudy(s).

Percentile_95(compute_percentile(p, 95)) :-
    TermProbability(t, p).

ans(t, p) :-
    TermProbability(t, p), percentile_95(p95), p > p95.
```

We can see in in table 1, on the right, how, extending the example proposed in Section 4.2 by using the knowledge stored in the CogAt ontology, we can filter out those terms that, being present in most neuroimaging studies, only add noise to the results. Therefore, we obtain a list of much more relevant results that are also more closely related to the general knowledge of the field of neuroscience. Solving this query takes approximately 6 seconds.

4.4 Retrieving information from related terms via the hierarchical structure of the ontology

We now show we can leverage the ontological knowledge provided by the International Organization for Biological Control (IOBC) to perform an analysis that includes terms related to our main term (*noxious* and *nociceptive* related to *pain*, in this example) without knowing them beforehand, enriching our analysis automatically. The IOBC ontology rewriting adds 11,102 formulas to our database.

```
RelatedTerm(term) :- term == "pain".
RelatedTerm(term) :-
    label(pain_entity, "pain"),
    related(pain_entity, subclass),
    altLabel(subclass, term).

FilteredBySynonym(t, s) :-
    TermInStudy(t, s), RelatedTerm(t).

Result(i, j, k, PROB) :-
    FocusReported(i, j, k, s)
    | SelectedStudy(s), FilteredBySynonym(t, s).

Percentile_95(compute_percentile(p, 95)) :-
    Result(i, j, k, p).

VoxelActivationImg(create_region_overlay(i, j, k, p)):-
    Result(i, j, k, p),
    Percentile_95(p95), p > p95.

ans(img) :- VoxelActivationImg(img).
```

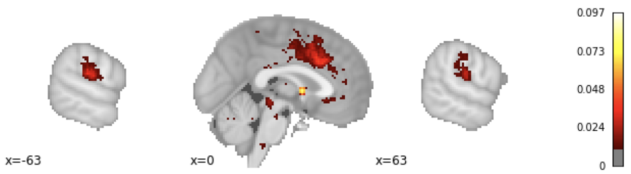

Figure 3: Resulting thresholded brain image from the NeuroLang use case showing the activations related to pain and its related terms derived from the IOBC ontology (noxious and nociceptive). Dorsal anterior cingulate cortex (x=0) and parietal regions are be active in articles mentioning pain and related words. agreeing with current knowledge in pain location (Lieberman and Eisenberger 2015).

Figure 3 provides a view of the results obtained from this example. In this case, the activations of Noxious and Nociceptive were also automatically included in the result, solving one of the current problems of NeuroSynth (the need to know all the terms you want to use beforehand). Solving this query takes approximately 55 seconds.

4.5 Segregation reverse inference query

This final use case shows how we can use negation and existential to express specificity. We pick the terms present in the CogAt ontology which are mentioned in documents reporting activations within the short insural gyri. Processing took 5.7 seconds. Results shown in table 2.

```
RegionActivated(r, s) :-
    VoxelByRegionDestrieux(i, j, k, r),
    FocusReported(i1, j1, k1, s).

ShortInsulaLabels("l_g_insular_short").
ShortInsulaLabels("r_g_insular_short").

ShortInsulaRegionsActive(s) :-
    RegionActivated(r, s), ShortInsulaLabels(r).
NonShortInsulaRegionsActive(s) :-
    RegionActivated(r, s), ~ShortInsulaLabels(r).

FilteredTerm(t, s) :- TermInStudy(s, t), label(uri, t).

TermProbability(t, PROB) :- FilteredTerm(t, s)
    | (ShortInsulaRegionsActive(s),
      ~NonShortInsulaRegionsActive(s),
      SelectedStudy(s)).

Percentile_75(compute_percentile(p, 75)) :-
    TermProbability(t, p).

ans(t, p) :-
    TermProbability(t, p), Percentile_75(p75), p > p75.
```

5 Discussion and Conclusion

In this paper, we presented a fragment of probabilistic Datalog+/– enriched with negation and aggregation, along with a scalable query resolution algorithm. The main goal of our specific approach is meta-analysis of neuroimaging data. Several different approaches to probabilistic Datalog+/– semantics and query resolution exist (Gottlob et al. 2013; Ceylan, Darwiche, and Van den Broeck 2021). Nonetheless, these do not incorporate aggregation, and the possibility of manipulating the probabilistic query results within the same language. These two features, as shown by our use-case analysis in Section 4, are fundamental traits required to provide a probabilistic logic programming language that can encode neuroimaging meta-analysis applications end-to-end.

term	prob
inhibition	0.250000
context	0.166667
memory	0.166667
perception	0.166667

Table 2: Terms mentioned in our segregation query in section 4.5. As expected short insula is specifically related most probably to inhibition tasks (Nieuwenhuys 2012).

The possibility of manipulating probabilities within the language comes at a great expense. After our PERs are computed, in Step 4 of Algorithm 1, our language allows handling probabilities as a standard float column. While this allows for analyses required by our target applications, it calls for disciplined programming from the user such that the manipulation of probabilities remains sound. Nonetheless, this gives our language great power; for instance, we can build probabilistic brain images, through aggregation, as shown in Sections 4.1–4.4; and compute the probability differences between two events, which we show in Section 4.5.

All these features allow us to go beyond current tools in meta analyses whose queries are based in propositional logic (Yarkoni et al. 2011; Laird et al. 2011) and harness the full power of the $\text{FO}^{\neg\exists}$ fragment, as well as open-world semantics, to express meta-analysis tasks in a sound, disciplined, and declarative manner. Furthermore, by using, as in Ceylan et al. (Ceylan, Darwiche, and Van den Broeck 2021), a lifted query processing approach when possible (see Algorithm 1, Step 3), we are able to process current meta-analytic datasets enriched with ontologies that are of considerable size, as described at the beginning of Section 4.

To conclude, we have shown that neuroimaging meta-analytic applications are an excellent real-world application for a language such as probabilistic Datalog+/–. By using probabilistic semantics that have recently converged from different probabilistic logic and open-world language approaches (Riguzzi 2008; Ceylan, Darwiche, and Van den Broeck 2021; Vennekens, Denecker, and Bruynooghe 2009), with open-world semantics (Calì, Gottlob, and Lukasiewicz 2012; Gottlob, Orsi, and Pieris 2014; Ceylan, Darwiche, and Van den Broeck 2021), and query resolution approaches (Dalvi and Suciu 2012; Ceylan, Darwiche, and Van den Broeck 2021; Vlasselaer et al.), we have produced a language that is ready to be used in neuroimaging applications.

Acknowledgement

This work was partially supported by the ERC-StG NeuroLang ID:757672. We are deeply thankful to the NiLearn community for the data ingestion and visualization

tools (Abraham et al. 2014).

References

Abiteboul, S.; Hull, R.; and Vianu, V. 1995. *Foundations of Databases*. Addison-Wesley. Addison Wesley.

Abraham, A.; Pedregosa, F.; Eickenberg, M.; Gervais, P.; Mueller, A.; Kossaifi, J.; Gramfort, A.; Thirion, B.; and Varoquaux, G. 2014. Machine learning for neuroimaging with scikit-learn. *Frontiers in Neuroinformatics* 8:14.

Baget, J.-F.; Leclère, M.; Mugnier, M.-L.; and Salvat, E. 2011. On rules with existential variables: Walking the decidability line. *Artificial Intelligence* 175(9-10):1620–1654.

Beeri, C., and Vardi, M. Y. 1981. The implication problem for data dependencies. In *Proc. of ICALP*, 73–85.

Calì, A.; Gottlob, G.; and Kifer, M. 2008. Taming the infinite chase: Query answering under expressive relational constraints. In *Proc. of KR*, 70–80.

Calì, A.; Gottlob, G.; and Lukasiewicz, T. 2012. A general datalog-based framework for tractable query answering over ontologies. *J. Web Semant.* 14:57–83.

Ceylan, İ. İ.; Darwiche, A.; and Van den Broeck, G. 2021. Open-world probabilistic databases: Semantics, algorithms, complexity. *Artificial Intelligence* 295:103474.

Dalvi, N., and Suciu, D. 2012. The dichotomy of probabilistic inference for unions of conjunctive queries. *Journal of the ACM* 59(6):1–87.

Dantsin, E.; Eiter, T.; Gottlob, G.; and Voronkov, A. 2001. Complexity and expressive power of logic programming. *ACM Computing Surveys (CSUR)* 33(3):374–425.

Destrieux, C.; Fischl, B.; Dale, A.; and Halgren, E. 2010. Automatic parcellation of human cortical gyri and sulci using standard anatomical nomenclature. 53(1):1–15.

Deutsch, A.; Nash, A.; and Remmel, J. B. 2008. The chase revisited. In *Proc. PODS-2008*, 149–158.

Fagin, R.; Kolaitis, P. G.; Miller, R. J.; and Popa, L. 2005a. Data exchange: Semantics and query answering. *Theor. Comput. Sci.* 336(1):89–124.

Fagin, R.; Kolaitis, P. G.; Miller, R. J.; and Popa, L. 2005b. Data exchange: semantics and query answering. *Theoretical Computer Science* 336(1):89–124. Database Theory.

Gottlob, G.; Lukasiewicz, T.; Martinez, M. V.; and Simari, G. I. 2013. Query answering under probabilistic uncertainty in datalog+/- ontologies. *Annals of Mathematics and Artificial Intelligence* 69(1):37–72.

Gottlob, G.; Orsi, G.; and Pieris, A. 2014. Query Rewriting and Optimization for Ontological Databases. *ACM Transactions on Database Systems* 39.

Laird, A. R.; Eickhoff, S. B.; Fox, P. M.; Uecker, A. M.; Ray, K. L.; Saenz, J. J.; McKay, D. R.; Bzdok, D.; Laird, R. W.; Robinson, J. L.; Turner, J. A.; Turkeltaub, P. E.; Lancaster, J. L.; and Fox, P. T. 2011. The BrainMap strategy for standardization, sharing, and meta-analysis of neuroimaging data. *BMC Research Notes* 4(1):349.

Lieberman, M. D., and Eisenberger, N. I. 2015. The dorsal anterior cingulate cortex is selective for pain: Results from large-scale reverse inference. *Proceedings of the National Academy of Sciences* 112(49):15250–15255.

Mesulam, M. M. 1998. From sensation to cognition. *Brain: A Journal of Neurology* 121 (Pt 6):1013–1052.

Nieuwenhuys, R. 2012. The insular cortex: A review. *Progress in Brain Research* 195:123–163.

Poldrack, R. A., and Yarkoni, T. 2016. From Brain Maps to Cognitive Ontologies: Informatics and the Search for Mental Structure. *Annual review of psychology* 67(1):587–612.

Poldrack, R. A.; Kittur, A.; Kalar, D.; Miller, E.; Seppa, C.; Gil, Y.; Parker, D. S.; Sabb, F. W.; and Bilder, R. M. 2011. The Cognitive Atlas: Toward a Knowledge Foundation for Cognitive Neuroscience. *Frontiers in Neuroinformatics* 5.

Riguzzi, F. 2006. ALLPAD: Approximate Learning of Logic Programs with Annotated Disjunctions. Technical Report CS-2006-01, University of Ferrara.

Riguzzi, F. 2008. ALLPAD: Approximate learning of logic programs with annotated disjunctions. *Machine Learning* 70(2-3):207–223.

Samartsidis, P.; Montagna, S.; Johnson, T. D.; and Nichols, T. E. 2017. The coordinate-based meta-analysis of neuroimaging data. *Statistical Science* 32(4).

Senellart, P. 2017. Provenance and Probabilities in Relational Databases: From Theory to Practice. *SIGMOD Record* 46(4):11.

Suciu, D.; Olteanu, D.; Ré, C.; and Koch, C. 2011. *Probabilistic Databases*. Morgan & Claypool.

Vennekens, J.; Denecker, M.; and Bruynooghe, M. 2009. CP-logic: A language of causal probabilistic events and its relation to logic programming. *Theory and Practice of Logic Programming* 9(3):245–308.

Vlasselaer, J.; Kimmig, A.; Dries, A.; Meert, W.; and Raedt, L. D. Knowledge Compilation and Weighted Model Counting for Inference in Probabilistic Logic Programs. In *The Workshops of the Thirtieth AAAI Conference on Artificial Intelligence Beyond NP*, 6.

Vlasselaer, J.; Renkens, J.; Van den Broeck, G.; and De Raedt, L. 2014. Compiling probabilistic logic programs into sentential decision diagrams. In *Workshop on Probabilistic Logic Programming (PLP)*.

Yarkoni, T.; Poldrack, R. A.; Nichols, T. E.; Van Essen, D. C.; and Wager, T. D. 2011. Large-scale automated synthesis of human functional neuroimaging data. *Nature Methods* 8(8):665–670.