

Probabilistic Answer Set Programming Driven Ranking of Dynamic Space-Time Belief Models

Julius Monsen^{1,3}, Jakob Suchan^{2,3}, and Mehul Bhatt^{1,3}

¹ Örebro University, Sweden

² Constructor University, Germany

³ CoDesign Lab EU., codesign-lab.org/cognitive-vision
cognitive.vision@codesign-lab.org

Abstract. A key challenge in embodied, inter(active) vision is reasoning over alternative hypotheses about the dynamics of perceived objects and events, be it for real-time or even offline interpretation. Towards this, we address the problem of generating and ranking grounded visuospatial hypotheses based on a semantically encoded notion of hypothesis preference. Driven by probabilistic Answer Set Programming (ASP), we propose a general framework for modeling and reasoning about diverse preference types tailored to visuospatial interpretation tasks. The effectiveness of our probabilistic visuospatial hypotheses ranking method is demonstrated and evaluated with a community benchmark of Multi-Object Tracking (MOT17), where modeling uncertainty and preference is critical for robust scene interpretation. Furthermore, practical examples also showcase how semantically driven reasoning with preferences can be effectively used in real-world visual sensemaking tasks.

Keywords: Probabilistic Answer Set Programming · Preferential Ranking · Visual Intelligence · Deep Semantics · Cognitive Vision

1 Introduction

The high-level, explainable interpretation of dynamic visual scenes—such as those encountered in autonomous driving, human-robot interaction etc—remains a fundamental challenge in AI and ML. While contemporary deep learning-driven visual computing excels at processing raw sensory input, it lacks mechanisms for abstract, context-aware, explainable reasoning and learning in a robust and formally verifiable manner. Likewise, relational AI and its practical declarative manifestations offer robust tools for structured / logic-based commonsense representation and reasoning, but typically struggle in their ability to handle large-scale, noisy, incomplete (dynamic) quantitative sensory data.

Embodied Visual Intelligence. In recent years, the integration of semantically rooted or knowledge-centric declarative AI methods for commonsense reasoning with deep learning-based visual computing techniques has emerged as a promising research direction aimed at achieving embodied (artificial) visual intelligence [5,23]. Here, visual intelligence corresponds to computational sensemaking under ecologically valid “*in-the-wild*” naturalistic settings where the crucial challenge is that of integrated explainable reasoning and learning about dynamic

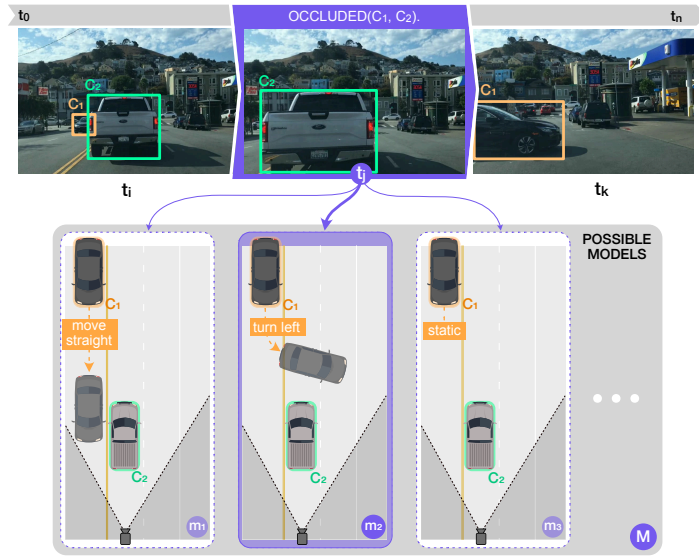


Fig. 1: **Dynamic Space-Time Belief Models.** An example illustrating the challenge of visual sensemaking. In this traffic scenario, a vehicle is occluded, creating ambiguity about the vehicle's possible trajectory. There are multiple plausible semantically distinct hypotheses—such as the car continuing straight (M1), turning left (M2), or remaining static (M3)—that may be derived and ranked according to, for instance, human safety preferences.

(inter)active multimodality [5]. To this end, neurosymbolic visual commonsense mechanisms for reasoning about spatio-temporal dynamics integrating semantically driven AI methods for commonsense knowledge representation and reasoning with quantitative visual computing techniques have emerged in particular [25,24]. The general position guiding these neurosymbolic mechanisms is that real-world sensemaking of visual stimuli (such as video) requires the ontological and computational interfacing of diverse multi-faceted methods in AI and ML. This position resonates directly with long-established, but often overlooked, convictions concerning realizing computational models of human-like cognitive vision; to quote Marr [20]:

» “The study of vision must [...] include not only the study of how to extract from images the various aspects of the world that are useful to us, but also an inquiry into the nature of the internal representations by which we capture this information and thus make it available as a basis for decisions about our thoughts and actions.”

This perspective underscores the necessity for *internal representations* that are not only *semantically descriptive* but are also *actionable*, i.e., capable of supporting inference, prediction, and semantic preference-driven selection (amongst other things). As an example, consider the example in Fig. 1, aimed at highlighting the need and value for high-level reasoning about space-time dynamics

under tight real-time visual sensemaking and control requirements in real-world conditions.

KEY CONTRIBUTIONS. This research builds on and methodologically advances an ongoing line of work on (computational) *cognitive vision* [4] aimed at integrating symbolic AI and sub-symbolic computer vision techniques. The key contributions presented in this paper are:

C1. Probabilistic Preferential Ranking: A systematic and general characterization of probabilistic preferential ranking of dynamic space-time belief models (temporally evolving, hypotheses about visual elements in a scene) that serves as the computational foundation enabling (re)ordering of neurosymbolically grounded perceptual beliefs emanating from low-level quantitative stimuli. Indeed, probabilistic preferential ranking naturally serves as a tool to not only encode semantic selection of preferences, but also to handle uncertainty that is inherent in quantitative data, thereby accounting for noisy or completely missing data, as well as errors in low-level visual processing methods.

C2. Declarative Neurosymbolic Model: Dynamic space-time preferential ranking is declaratively implemented as a general, neurosymbolic method driven by an integration of probabilistic Answer Set Programming (ASP) [8,14] and low-level visual computing mechanisms (e.g., for object detection). The implemented method is primarily, but not exclusively, driven by supporting the formalization of *policies* that encode both domain-independent and dependent factors such as: (a) Constraints relevant to generally applicable commonsense laws pertaining to spatio-temporal dynamics as encountered in everyday “in-the-wild” embodied interaction settings; (b) Heuristics representing high-level cognitive preferences characterizing human/cognitive mental model selection mechanisms as part of abductively computing alternate or counterfactual hypotheses computation; and (c) Contextual knowledge and other considerations/requirements resulting from the specifics of an application domain/use-case.

Our overall goal is to develop systematically formalized and implemented modular, extensible foundational mechanisms that enable robust neurosymbolic visual sensemaking in real-world problem settings. Therefore, we also showcase and evaluate the practical utility of our proposed probabilistic visuospatial hypotheses ranking methodology with MOT17⁴, a computer vision, community established, benchmark focussing on Multi-Object Tracking (MOT) [21].

2 Background: Declarative Visuospatial Commonsense

An established line of research on computational commonsense representation of space and time is that qualitative representations of spatial, temporal, and spatio-temporal relations (e.g. *front_right*, *touching*, *during*, *approaching*) can effectively map between high-level semantic categories and low-level quantitative data. For instance, abstracting from metric details of the physical world, such relational knowledge can capture 4D space-time regions with both spatial and temporal components [9] for modeling object interactions [26]. The influence

⁴ MOT17 Challenge: <https://motchallenge.net/data/MOT17/>

of (spatio-linguistic) “path & motion predicates” for grounding complex quantitative data is also well-recognized [19]. Such commonsense characterizations are a promising abstraction mechanism for semantic grounding and automated reasoning about embodied multimodal interaction encompassing space, actions, events, change and motion [3].

Commonsense representations of space, time, and motion naturally align and integrate with knowledge representation (KR) frameworks that support expressive, high-level modeling and reasoning. In this context, ASP [6,16,7] has emerged as a particularly suitable paradigm for formalizing and operationalizing such dynamic, relational knowledge about space, time, and motion [26,24,22]. This is by virtue of the fact that ASP is also one of the most scalable and actively researched methods for non-monotonic reasoning (e.g., required for belief revision), an aspect critical to the technical agenda of this research. Several extensions to the core ASP framework have accrued recently, e.g., functional ASP for modeling specialized continuous domains by satisfiability modulo theory (SMT) integration [13,1], probabilistic ASP [15,8], inductive learning [12], spatial reasoning [26], neurosymbolic integration [28,25].

To reason about uncertainty, which is inherent in visual interpretation tasks, we employ ASP, realized through Plingo [8], a probabilistic extension of Clingo [7]. Plingo is based on the $Lpmln\pm$ formalism—a variant of the probabilistic language $Lpmln$ [14]—that merges declarative logic programming with probabilistic reasoning. This framework is particularly well-suited for our purposes as it provides a well-integrated method for combining hard constraints with probabilistic preferences enabled by soft constraints, i.e. rules that do not strictly eliminate candidate solutions, but instead weight them probabilistically. This allows for representing and reasoning about uncertain knowledge in a declarative manner. Specifically, a Plingo program consists of two parts: (1) A hard part, written as normal ASP rules and weak constraints at priority levels greater than 0; and (2) A soft part, written as ASP weak constraints on the form $:\sim F[w, l]$ (where F is a formula, w is a real number weight, and l is a nonnegative integer) at priority level 0, that assigns weights to the model, and thereby induces a probability distribution over all stable models.

More formally, we can express the semantics of $Lpmln\pm$ (and in extension Plingo) in terms of a logic program $\Pi = \Pi^{hard} \cup \Pi^{soft}$ where Π^{hard} is the hard part and Π^{soft} is the soft part. Let $SSM(\Pi)$ denote the set of soft stable models of Π . An interpretation (a set of atoms) X is included in this set if it is a stable model of the program formed by the rules in Π that X satisfies (viewed as unweighted rules). To define the probability of such a model, we first define its weight. Let Π_X be the set of weighted rules from Π satisfied by X . The weight of the interpretation, $W_\Pi(X)$, is then the total weight of this subprogram $TW(\Pi_X) = exp(\sum_{w: F \in \Pi_X} w)$, as seen in Equation 1. Finally, the probability $P_\Pi(X)$ of a model $X \in SSM(\Pi)$ is determined by normalizing the weights over all such optimal stable models $W_\Pi(Y)$, as shown in Equation 2. Here, the limit $\alpha \rightarrow \infty$ in the probability formula enforces the infinite weight of hard rules, ensuring that any model violating them receives zero probability. X is then a probabilistic stable model of Π if $P_\Pi(X) \neq 0$.

$$W_{\Pi}(X) = \begin{cases} TW(\Pi_X) & X \in SSM(\Pi) \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

$$P_{\Pi}(A) = \lim_{\alpha \rightarrow \infty} \frac{W_{\Pi}(X)}{\sum_{Y \in SSM(\Pi)} W_{\Pi}(Y)} \quad (2)$$

When comparing two stable models X and Y of Π^{hard} , X is preferred over Y (with respect to the weak constraints Π^{soft}) if there is a level l where X has a lower cost than Y , and at every higher level $l' > l$, their costs are the same [8]. An optimal stable model of Π is a stable model X for which there is no stable model Y of Π^{hard} such that Y is preferred over X with respect to Π^{soft} . In Section 3, we will show how such preferences can be specified in a declarative manner within Plingo, and how they can be used to rank different interpretations of a visual scene.

3 Preferential Ranking of Dynamic Space-time Belief Models

Within the context of this work, we focus on the problem of ranking dynamic space-time belief models, i.e. stable models of a logic program that represent the possible interpretations of the scene and the objects in it over time. To rank these models effectively, it is necessary to define preferences related to key aspects of the visual domain.

3.1 Space and Motion Preliminaries

Our framework adopts a formalization of space, motion, objects, and events inspired by prior work, particularly the approach by [25,24]. At the core of this formalization are two key domains: the spatio-temporal domain Σ_{st} , representing the geometric and temporal aspects of visual objects, and the dynamic domain Σ_{dyn} , capturing higher-level event-based descriptions of change over time.

In the spatio-temporal domain Σ_{st} , we consider a set of domain objects $\mathcal{O} = o_1, \dots, o_n$, which correspond to entities in the visual scene, e.g. cars and pedestrians. Each object o_i is represented as a spatial entity $\varepsilon_i \in \mathcal{E}$, where $\mathcal{E}$ may include geometric forms like points, lines, or polygons in 2D or 3D space. Temporal dynamics are encoded via a set of discrete time points $\mathcal{T} = \{t_1, \dots, t_m\}$, and object motion is captured through trajectories $\mathcal{MT}_{o_i} = (\varepsilon_{t_s}, \dots, \varepsilon_{t_e})$, which describe how spatial entities evolve from start time t_s to end time t_e . Additionally, spatio-temporal relations $\mathcal{R}$ define the qualitative relationships between objects across space and time.

The dynamic domain Σ_{dyn} enriches this structural representation with semantic content in the form of fluents $\Phi = \phi_1, \dots, \phi_n$ and events $\Theta = \theta_1, \dots, \theta_n$, which respectively represent the state-dependent properties of the world and the occurrences that alter those states. We say that an event $\theta_i \in \Theta$ is considered possible only if a set of necessary conditions c_j are satisfied. We express this as: $Poss(\theta_i) \leftrightarrow \bigwedge_j c_j$. To reason about these dynamics, we use an ASP-based formalization [18] of the event calculus [10], wherein axioms such as `occurs_at`(θ, t)

assert that an event θ occurs at time t , and `holds_at`(ϕ, v, t) asserts that a fluent ϕ takes on a value v at time t .

3.2 Preferences in Dynamic Space-time Belief Models

We define Σ_{prf} as the domain preferences that can be used to rank dynamic space-time belief models. Σ_{prf} consist of preferences $\mathcal{P}$ that are defined over the events Θ in the dynamic domain Σ_{dyn} where each event $\theta_i \in \Theta$ may involve one or more objects ($\mathcal{O}_{\theta_i}$).

Our notion of preference is defined as a tuple consisting of a set of preference functions (PF_{θ_i}), and a corresponding set of weights ($w_{\theta_i,1}$). The preference function for an event $\theta_i \in \Theta$ is defined over a finite number of objects involved in the event $\{o_1, \dots, o_n\} \in \mathcal{O}_{\theta_i}$ as follows:

$$PF_{\theta_i} : \mathcal{O}_{\theta_i} \times \dots \times \mathcal{O}_{\theta_i} \rightarrow \mathbb{R} \quad (3)$$

This results in a set of *preference features* $\mathcal{F}_{\theta_i} = \{f_{\theta_i,1}, \dots, f_{\theta_i,m}\}$ associated with θ_i . These preferences can then be used to guide the ranking of competing space-time belief models by quantifying the desirability of including the event θ_i under the given domain preferences. The preferences may belong to any of the following categories, where they differ in terms of the feature set the preference function is defined over:

- $\mathcal{P}_{\text{geom}}$ denotes preferences based on the geometric properties of the objects in the scene and their dynamics over time. This mostly concerns the spatial entities $\mathcal{E}$ and the motion trajectories $\mathcal{MT}_{o_i}$ of the objects o_i in the scene. For example, we can define preferences involving distances between objects, the size of the objects, the speed of the objects within a certain time interval, the direction of motion, or generally how geometric objects should behave by the laws of physics.
- $\mathcal{P}_{\text{vis}}$ denotes preferences based on features extracted from visual data with neural networks. In principle, there is no limitation to what this kind of features can be used to define these preferences. For example, we can define preferences based on visual appearance features (e.g. re-identification features that compute an embedding similarity between objects), object type classification, the 3D depth of the objects, object pose and orientation as well as aspects in the general context of the scene.
- $\mathcal{P}_{\text{sem}}$ denotes preferences based on semantic properties at the object, scene, and conceptual levels. At the object level, these include type, color, affordances, and relationships (e.g., interactions, and hierarchies). Scene-level preferences depend on context, such as layout (indoor, outdoor, traffic) and associated knowledge (e.g., traffic rules). Conceptual-level preferences reflect higher-order aspects like regulations, safety, and norms—for example, obeying traffic signals, maintaining safe distances, or adjusting behavior in school zones. These preferences guide actions toward, for instance, legal and safety-compliant outcomes.
- $\mathcal{P}_{\text{event}}$ denotes preferences that are based on the events that can occur in the scene and the causal relationships between them. This is dependent on the dynamic domain Σ_{dyn} . For example, we can define preferences stating that if an object (e.g. a vase) is hit by another object (e.g. a ball), it likely affects the

state of one or both objects (e.g. the vase is broken) depending on the the type of objects involved in the event and the motion of the objects.

3.3 Weighting Preference Features

We pair the preference features with weights that quantify the importance of each feature for an event. We define a set of weights as $\mathcal{W}_{\theta_i} = \{w_{\theta_i,1}, \dots, w_{\theta_i,m}\}$ for each event $\theta_i \in \Theta$ and include it in the preference domain Σ_{prf} . The selected events with their associated features and their weights then influence the likelihood of models where particular event patterns occur and ultimately decide if an event θ_i should be preferred or not in the final model X of the logic program Π . In our ASP framework for ranking dynamic spatio-temporal belief models, preferences are encoded using the special theory atom `&weight(W)` in Plingo. This atom is a special construct that directly associates a numerical weight with the head of the rule it appears in. When the body of a rule is satisfied, its associated `&weight(W)` value is included in the sum of weights used to calculate the total model weight, $W_{\Pi}(X)$, as described in Section 2. In this context, a higher weight signifies a stronger preference, contributing positively to the model's overall likelihood. For example, we can write:

```
occurs_at(turn_left(car1), curr_time) :- &weight("0.6").
```

to indicate that the event `turn_left(car1)` is preferred with a weight of 0.6. Then if we have another possibility `turn_right(car1)` with a weight of 0.2, and only one action can occur at each timestep, the model with the first event `turn_left(car1)` will be preferred. As an example, a preference may be encoded as follows (note that the `@` symbol denotes a call to an external function outside of the answer set solver):

```
w(preferred(turn_left(car1)), W) :- possible(turn_left(car1)),
    F = @compute_pref_feature(turn_left(car1)), W = @weight((F), (w)).
```

The expression `W = @weight((F), (w)).` computes the weighted feature value $f_{turn_left(car1),1} \cdot w_{turn_left(car1),1}$, where $w_{turn_left(car1),1} \in [0, 1]$. This allows us to express both soft constraints and graded preferences in a fine-grained and interpretable manner. However, Plingo does not natively support computing weights from dynamic expressions within a stable model without modifying the encoding itself at every solver call. To address this, we introduce a preprocessing step that transforms rules (as exemplified in the code snippet above) into explicit weight declarations using Plingo's theory atom by first grounding the `w(Preference, W)` predicates (during which external functions are called) and then re-writing the resulting rules into the following form such that the Plingo solver can process them:

```
preferred(turn_left(car1), f) :- &weight("0.6").
```

Additionally, we ensure that the preference weights associated with an event are only considered if that event is selected in a given stable model. If an event is not part of the selected model, then its associated feature weights are excluded from

the model’s cost computation. This selective inclusion guarantees that the total cost reflects only the features and preferences relevant to the actual interpretation represented by the model. The cost of a dynamic space-time belief model X is calculated as in Equation 4 by summing the contributions of all features f —over all preference categories— associated with each selected event θ_i in the model, weighted by their corresponding weights.

$$\text{Score}(X) = \sum_{\theta_i \in \Theta_X} \sum_{f_{\theta_{ij}} \in \mathcal{F}_{\theta_i}} w_{\theta_{ij}} \cdot f_{\theta_{ij}}(\mathcal{O}_{\theta_i}) \quad (4)$$

Overall, this weighting mechanism provides a flexible and adaptive way of integrating low-level heterogeneous signals and high-level domain-specific knowledge into the model ranking process, enabling robust interpretation in dynamic visual environments. Taken together, the representational ontology can be summarised as a unified domain Σ defined in (5):

$$\Sigma \equiv_{def} \Sigma_{dyn} \langle \Phi, \Theta \rangle \cup \Sigma_{st} \langle \mathcal{O}, \mathcal{E}, \mathcal{T}, \mathcal{MT}, \mathcal{R} \rangle \cup \Sigma_{prf} \langle \mathcal{P}, \mathcal{W} \rangle \quad (5)$$

Learning Weights for Preferences: The preference weights roughly assert how important each rule is in deriving a stable model [15]. For simple programs, the weights can be manually specified. However, systematically assigning weights in complex and dynamic domains such as visual sensemaking can be challenging. Above all, the interdependencies between rules can make it difficult to determine the relative importance of individual weights.

In [15], the authors proposed a method for learning the weights of the rules from the observed data. Following similar principles as those used for Maximum Likelihood Estimation (MLE) in the practice of machine learning, they used Markov Chain Monte Carlo (MCMC) sampling, adapted to use ASP solvers for performing the sampling to learn the weights of the rules in a logic program. In line with this approach, we adopt a simpler yet effective strategy for weight selection based on gradient-free random search in the space of possible weights. While more sophisticated optimization techniques exist and can be implemented in the future, we find that random search is sufficient and robust for exploring the weight space in our case study without making strong assumptions about the continuity or smoothness of the underlying cost landscape.

4 Visual Interpretation with Moving Objects

To demonstrate applicability and to evaluate practical performance of our framework, we present a case study in Multi-Object Tracking (MOT) [21]. We adopt the widely used Tracking-by-Detection paradigm, where the goal is to associate visual observations (detections) with objects tracks over time, maintaining object identities and capturing their dynamic interactions. The association task can be framed as a combinatorial optimization problem, where the objective is to assign detections to object tracks in a way that minimizes a cost function. This cost is typically informed by distance to predicted object locations and feature similarities such as appearance or motion. Algorithms such as the Hungarian algorithm

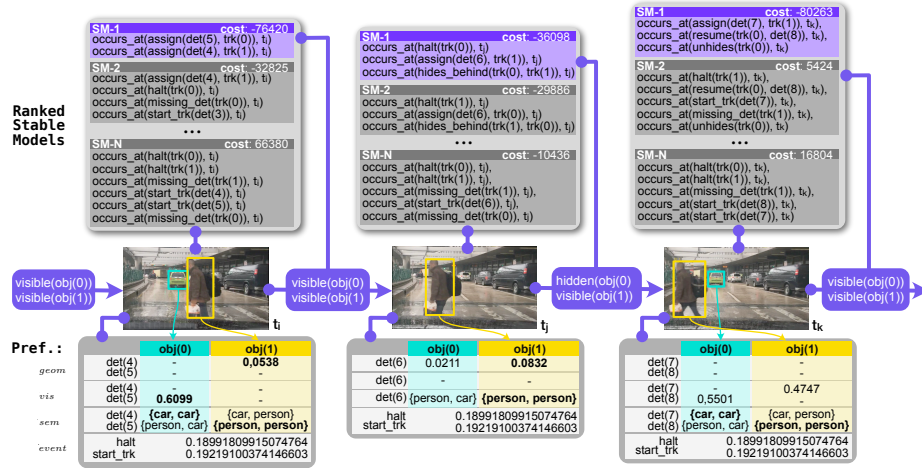


Fig. 2: **An example of the ranking process.** The figure illustrates how the preferential ranking process creates a set of alternative belief models with different costs (top). Each model represents the scene dynamics, including object interactions and events. In this case a person is walking in front of a car, then the car is occluded and then reappears once the person has passed. At the bottom, we can observe the preferences and their weighted features which are considered in the solving process and ultimately reflected in the costs.

[11] are commonly used to solve this. A well-known approach based on this technique is SORT [2], which uses a Kalman filter to predict object positions and applies the Hungarian algorithm to associate observations with these predictions. However, low-level matching is typically insufficient. High-level reasoning about scene dynamics can help disambiguate uncertain associations and lead to a more coherent understanding of the scene. Preliminary work has shown promise in this direction. For example, a method that jointly addresses observation-object association and explains scene dynamics through high-level events has been proposed in [24]. In this context, the association task can be reframed as a probabilistic reasoning problem. The objective becomes generating and evaluating multiple plausible hypotheses about the scene and object dynamics, and then ranking these hypotheses based on weighted preferences. This is where our framework for ranking dynamic space-time belief models becomes relevant.

4.1 Use-Case: Preferences in Multi-Object Tracking

Figure 2 presents an example of the key aspects of the proposed framework. It shows how preferences give rise to multiple models, each representing a different hypothesis about the underlying events involving the objects and observations in the scene. Through the application of preferences and their weighted features, the framework computes a ranking over these models and selects the most plausible interpretation based on the cost. To systematically evaluate the contributions of different preference types, we evaluate three tracker variants. These build incrementally on each other: PRF-SORT, which uses purely geometric pref-

erences akin to SORT; PRF-DeepSORT, which adds appearance-based features; and finally PRF-Track, which further integrates high-level, event-based abductive reasoning. The following subsections will detail the implementation of PRF-Track, as it showcases all facets of our framework, while the other two variants will serve as crucial baselines in the empirical evaluation. Algorithm 1 provides a high-level overview of our approach for ranking dynamic space-time belief models in the context of MOT.

Algorithm 1: Preferential Ranking of Space-Time Belief Models ($\mathcal{V}, \Sigma$)

Data: Visual imagery ($\mathcal{V}$) and background knowledge $\Sigma \equiv_{def} \Sigma_{dyn} \cup \Sigma_{st} \cup \Sigma_{prf}$
Result: Visual Explanations ($\mathcal{EX}\mathcal{P}$)

```

1  $\mathcal{MT} \leftarrow \emptyset, \mathcal{H}^{events} \leftarrow \emptyset$ 
2 foreach  $t \in T$  do
3    $\mathcal{VO}_t \leftarrow \text{observe}(\mathcal{V}_t)$ ;
4   foreach  $trk \in \mathcal{MT}$  do
5      $trk_{pred} \leftarrow \text{kalmán\_predict}(trk)$ ;
6    $\mathcal{ST}_t \leftarrow \mathcal{VO}_t \cup \mathcal{MT} \cup trk_{pred}$ 
7   // Preference extraction
8    $\mathcal{PRFS} \leftarrow \emptyset$ ;
9   foreach  $\theta_i \in \Theta$  do
10    foreach  $PF_{\theta_i} \in \mathcal{P}$  do
11       $f \leftarrow PF_{\theta_i}(\mathcal{O}_{\theta_i}, \dots, \mathcal{O}_{\theta_i})$ ;
12      if  $\text{possible}(\theta_i, f)$  then
13        // Apply weights to features
14         $\mathcal{PRF}' \leftarrow f \times W_{\theta_{ij}}$ ;
15         $\mathcal{PRFS} \leftarrow \mathcal{PRFS} \cup \mathcal{PRF}'$ ;
16    // ASP solving and ranking
17     $\text{Models} \leftarrow \text{plingo\_solve}(\mathcal{PRFS}, \mathcal{V}, \mathcal{ST}_t, \Sigma)$  where  $\text{Models} = \{(X_1, \text{Cost}_1), \dots\}$ ;
18     $\mathcal{H}_t^{events} \leftarrow \text{preferred\_model}(\text{Models})$ ;
19     $\mathcal{H}^{events} \leftarrow \mathcal{H}^{events} \cup \mathcal{H}_t^{events}$ 
20   $\mathcal{MT} \leftarrow \text{update}(\mathcal{MT}, \mathcal{VO}_t, \mathcal{H}^{events})$ 
21 return  $\mathcal{EX}\mathcal{P} \leftarrow \langle \mathcal{H}^{events}, \mathcal{MT} \rangle$ 

```

I. Domain Predicates and Facts: In line with the formalism presented in Section 3.1, the inputs at each timepoint t are a set of visual observations (detections) $\mathcal{VO}_t = \{\varepsilon_{obs_1}, \dots, \varepsilon_{obs_n}\}$ represented as 2D bounding boxes $\mathcal{E}$, and a set of existing object tracks $\mathcal{MT}_{t-1} = \{\mathcal{MT}_{o_1}, \dots, \mathcal{MT}_{o_n}\}$. Each detection is characterized by its class (e.g., pedestrian) and a confidence score in the range $[0, 100]$ indicating the likelihood of the object’s presence and its classification. In our ASP encoding, these inputs are represented as a set of facts: **det**(ID, Type, Conf) for detections and **trk**(ID, Type) for tracks. The overall task is to find the most plausible model X consisting of a set of tracking events from Θ that explain new detections in light of currently tracked objects.

II. Generating Event Possibilities: In MOT, we have the relevant events $\Theta = \{assign, halt, start, resume\}$. Each event represents a hypothesis about the state transition of an object or observation in the scene: *start* indicates that an observation should initiate a new object track; *assign* denotes an association between an observation and an existing object track; *halt* signals that an object is temporarily not visible (due to occlusion or missing detections); *resume* suggests that an observation is linked to an object previously halted. The next step is to define the conditions under which these events can occur. We use `possible(Event)` to encode the preconditions for event feasibility.

For example, an *assign* event is defined by the following rules which check for either a high Re-identification (ReID) similarity or, failing that, a sufficient Intersection over Union (IoU) overlap:

```
possible(assign_reid(Det, Trk)) :- det(Det, Det_Type, Conf), Conf > 50, trk(Trk, Trk_Type),
    not holds_at(hidden(Trk), curr_time), not holds_at(clipped(Trk), curr_time),
    @reid_similarity(Trk, Det) >= reid_threshold.

possible(assign_iou(Det, Trk)) :- det(Det, Det_Type, Conf), Conf > 50, trk(Trk, Trk_Type),
    not holds_at(hidden(Trk), curr_time), not holds_at(clipped(Trk), curr_time),
    @reid_similarity(Trk, Det) < reid_threshold, @iou(Trk, Det) >= iou_threshold.
```

These rules state that a detection is a possible match for a track if (1) the detection has a confidence score greater than 50, (2) the track is neither hidden nor clipped due to another track occluding it or missing detections, and (3) the re-identification (ReID) cosine similarity between the detection and the track is above a defined threshold. We set the ReID and the IoU thresholds to 0.3 and 0.5, respectively.

Similar logic applies to other events. A *resume* event requires the track to have been previously hidden or clipped in addition to similar matching criteria as *assign*, and a *start* event is possible for any new detection with sufficiently high confidence. To prevent spurious tracks from noisy detections we follow common practice and introduce a constraint that a new object is only confirmed after being associated with a detection for three consecutive frames. Finally, any active track is allowed to *halt*, and will be removed from the scene if it remains halted beyond a maximum age threshold. We use a maximum age threshold of 1 frame for PRF-SORT and 30 frames for PRF-DeepSORT and PRF-Track.

III. Forming Plausible Hypotheses: Once the space of possible events is defined, we use choice rules and integrity constraints to generate concrete hypotheses about the events in the scene at timepoint t . Crucially, for each detection and track, exactly one event from the set of possibilities must be chosen. This forms the basis for generating candidate tracking models, where each stable model corresponds to a different combination of plausible events that explains the current state of the scene.

Furthermore, we incorporate abductive reasoning to infer plausible causes behind *halt* and *resume* events. If an active track becomes halted, it must be explained by either (i) the object being occluded by another object, identified via spatial overlap (using `@overlapping_top(Trk)`); or (ii) the object not being overlapped by any detection, suggesting missing observations. Similarly, if a previously halted

object reappears, the model must account for this either by determining that it has become visible again after being occluded, or by identifying it as recovering from a clipping scenario.

```

{occurs_at(hides_behind(Trk, Trk2), curr_time):
  Trk2 = @overlapping_top(Trk);
  occurs_at(missing_detections(Trk), curr_time):
    @not_overlapping_top(Trk)
} = 1 :- occurs_at(halt(Trk), curr_time), trk_state(Trk, active).

{occurs_at(unhides_from_behind(Trk, Trk2), curr_time):
  holds_at(hidden_by(Trk, Trk2), curr_time), not holds_at(hidden(Trk2), curr_time),
  @predicted_pos_within_distance(Trk, Det);
  occurs_at(recover(Trk), curr_time):
    holds_at(clipped(Trk), curr_time)
} = 1 :- occurs_at(resume(Trk, Det), curr_time).

```

This abductive layer serves two complementary purposes: (i) enhancing interpretability by offering human-readable explanations for events such as occlusion or reappearance, and (ii) acting as a semantic filter that prunes inconsistent or unsupported event hypotheses.

IV. Ranking Hypotheses with Preferences: To rank the competing hypotheses (stable models), we encode preferences as soft constraints. For MOT, these preferences are based on established features from the literature, such as ReID similarity and IoU overlap. For instance, in the case of the *assign* event, we can use ReID similarity and IoU to quantify how well a detection aligns with an existing object track. We first define ASP rules that link possible events to a weighted feature value.

```

preference(assign_iou(Det, Trk), W) :- possible(assign_iou(Det, Trk)),
  I = @iou(Trk, Det), W = @weight((I), (w1)).

preference(assign_reid(Det, Trk), W) :- possible(assign_reid(Det, Trk)),
  R = @reid_cosine_similarity(Trk, Det), W = @weight((R), (w2)).

```

As described in Section 3.3, our preprocessing step transforms these into grounded weighted rules that Plingo use for solving. For example, given a computed ReID similarity of 0.85, it is multiplied by a learned importance weight for the corresponding ReID feature, say $w_{\text{reid}} = 0.4$, yielding the following rule:

```

preference(assign(Det, Trk), reid) :- &weight("0.34").

```

This weight is then used directly by the Plingo solver to bias the selection of one assignment over another and ultimately rank the models. For the events *halt* and *start*, we do not compute feature scores, as these events typically serve as fallback options when other, more informative associations are not possible. Nevertheless, we still assign them preference weights to ensure they are incorporated into the ranking process. These static preferences help balance model selection by penalizing overuse of fallback events while still allowing them when necessary to preserve continuity or recover from uncertainty. However, for the *start* event, we associate slightly different weights depending on whether the concerned detection can be associated with an existing object track, through

assign or *resume* events. This completes the ranking procedure for dynamic space-time belief models in the context of MOT, enabling interpretable and flexible integration of data-driven signals and high-level reasoning.

V. Learning Preference Weights for Tracking: The final step is to determine the importance weights (e.g., w_1 , w_2 above). Effective weight selection in MOT requires a balance between competing objectives: associating observations accurately, avoiding identity switches, handling occlusions, and maintaining continuity in object trajectories. As discussed in Section 3.3, we use a random search strategy to explore the weight space for the preferences. We do this by sampling random weight values for each preference feature, within the defined range $[0, 1]$, and evaluating the tracking performance (for n -sized chunks of frames with n increasing over time to balancing local consistency and long-term identity preservation), optimizing for HOTA score [17] on the MOT17 training split. Once learned, we use these fixed preference weights during the final evaluation on the validation split of MOT17 as described in the following section.

4.2 Empirical Evaluation

We empirically evaluate our declarative tracking framework in terms of both tracking performance and computational efficiency.

I. Tracking Performance: We evaluate the proposed framework for preferential ranking on the validation split of MOT17 [21], a widely used benchmark for MOT. Table 1 summarizes performance across standard tracking metrics: HOTA [17], IDF1, MOTA, association accuracy (AssA), detection accuracy (DetA), and the total number of identity switches (IDs). Here, HOTA serves as the main performance indicator, as it holistically balances detection quality (DetA) with association consistency (AssA). IDF1 is particularly sensitive to long-term trajectory correctness, and the traditional MOTA score aggregates false positives, false negatives, and identity switches.

Method	HOTA	IDF1	MOTA	AssA	DetA	IDs
SORT [2]	59.6	60.4	58.0	59.8	59.5	398
DeepSORT [27]	67.6	70.0	62.7	70.6	64.7	250
PRF-SORT	59.9	60.4	58.7	59.6	60.3	423
PRF-DeepSORT	67.8	70.5	62.0	72.2	63.8	298
PRF-Track	68.0	70.4	62.0	72.6	63.8	268

Table 1: **Tracking performance on the MOT17 validation split.** Compared are original SORT & DeepSORT with our approach, PRF-SORT, PRF-DeepSORT, PRF-Track.

We compare the original SORT and DeepSORT [27] methods with our declarative counterparts—PRF-SORT, PRF-DeepSORT—and our full implementation PRF-Track, which integrates abductive reasoning and event-driven preferences:

- PRF-SORT: Our declarative SORT-inspired implementation in the developed ASP-driven framework. While it mirrors the original SORT in terms of features

used for matching, it replaces the Hungarian algorithm with an abductive and declarative ranking approach. Its performance closely matches that of the original SORT, demonstrating that ASP can effectively capture the same geometric matching logic.

- **PRF-DeepSORT:** Our declarative DeepSORT-inspired implementation. It leverages the same appearance-based features as DeepSORT but within the preference ranking framework. It slightly improves over DeepSORT in terms of HOTA and AssA, showing the benefit of structured preference-guided reasoning. However, it incurs a minor increase in ID switches.
- **PRF-Track:** This is our complete system, integrating abductive reasoning, as done in the above two methods, with the addition of high-level event-based preferences. It achieves the highest HOTA (68.0) and AssA (72.6), with competitive MOTA and IDF1 scores, and relatively few identity switches. These results illustrate the benefit of preference-guided model selection grounded in interpretable features and event structures.

These results demonstrate that our declarative and logic-based approach is competitive with conventional trackers while offering added benefits in terms of interpretability, modularity, and the ability to embed domain knowledge and preference trade-offs directly into the tracking logic. Moreover, by explicitly modeling events such as occlusion and recovery, our system yields more coherent and explainable tracking behavior in complex scenes.

II. Efficiency and Scalability: Beyond tracking accuracy, we evaluate the computational efficiency and scalability of our declarative trackers. We do not directly compare runtimes with the original implementations of SORT and DeepSORT. For context, trackers like SORT can often process several hundred frames per second. However, a direct comparison is challenging due to fundamental differences in implementation, and our goal is not to compete on this metric. Instead, our analysis aims to demonstrate that our declarative approach remains computationally feasible for real-world scenarios while offering gains in transparency and explainability. We analyze runtime performance for the three tracker variants by measuring: (1) average preprocessing time (initial grounding) per sequence, (2) average Plingo runtime (grounding + solving) per sequence and per frame, (3) number of event possibilities per frame⁵, (4) the size of the grounded logic program (atoms), and (5) models enumerated.

Table 2 reports these averages over sequences and frames respectively. As expected, expressiveness correlates with slight computational overhead. PRF-SORT remains the most lightweight, with an average solve time of 0.0070 s per frame, owing to its simple geometric matching logic. PRF-DeepSORT, which integrates ReID features, shows a slight increase in average grounding + solving time (0.0092 s), reflecting the additional reasoning over visual similarity metrics. PRF-Track, which further incorporates preference-guided abductive reasoning over events, introduces a marginally higher logical complexity (indicated by the average number of atoms per frame). However, with an average of 0.0164 s per frame, it still operates within real-time constraints for most practical applications.

⁵ At a minimum, equal to the number of objects and detections per frame.

Tracker	/Sequence (s)		/Frame				
	Preproc.	Plingo	Plingo (s)	Poss.	Atoms	Solving (ms)	Enum.
PRF-SORT	1.47	3.00	0.0070	42.4	457.2	0.039	2.25
PRF-DeepSORT	6.77	3.86	0.0092	48.8	1065.6	0.088	6.04
PRF-Track	7.95	6.74	0.0164	48.9	1499.2	0.093	5.81

Table 2: **Runtime statistics.** Average runtime and logical complexity for each tracker variant. Sequence-level times are shown separately from per-frame averages.

To better understand how runtime scales with input complexity, Figure 3 show Plingo time plotted against the number of ground atoms and the number of event possibilities, respectively. Across all methods, solving time exhibits an early exponential growth trend as the number of atoms increases. This reflects the greater grounding burden and increased model enumeration under more expressive reasoning conditions. We observe that the number of possible events correlates with a modest increase in Plingo time, and that PRF-Track separates from PRF-SORT and PRF-DeepSORT, being slightly slower due to the additional complexity induced by event-based preferences.

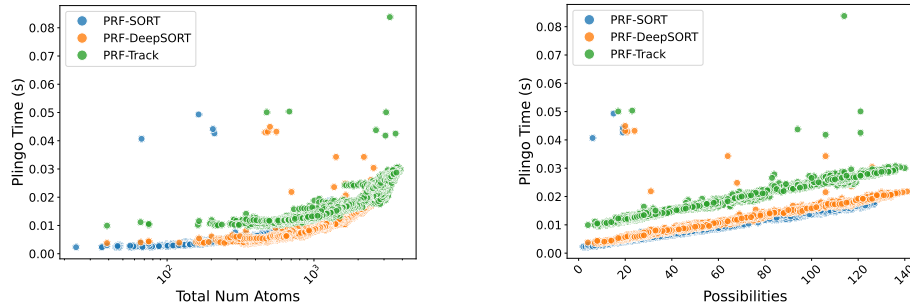


Fig. 3: **Comparison of Plingo execution time under different metrics.** On the y-axis, we show the Plingo time in seconds for each frame, while the x-axis shows the number of atoms in the logic program (left) and the number of event possibilities (right).

Finally, we note that PRF-DeepSORT and PRF-Track, both using appearance features, exhibit higher variance in solve time per frame with standard deviations of 0.105 ms and 0.097 ms, respectively. In contrast, the simpler PRF-SORT shows a more stable performance with a standard deviation of 0.015ms. Additionally, the results suggest that most of the computational overhead stems from the grounding stage, offering future optimization opportunities.

5 Summary and Outlook

Building on recent developments in neurosymbolic computational cognitive vision using Answer Set Programming (ASP) [23], we develop a declarative system

for probabilistically ranking dynamic space-time belief models. Driven by probabilistic extensions of ASP as realized in Plingo, our developed method is conceptually expressive, modular, and enables integration of visual, geometric, semantic, and event-based preference reasoning in a unified declaratively implemented formalism. At its core, our system generates and ranks dynamic scene interpretations, materialized as *stable models*, based on relational, feature-weighted preferences reflecting human-interpretable statistical features. This declarative approach enables seamless integration of diverse preference types aimed at handling ambiguity, partial observability, domain knowledge, or other types of semantic constraints (e.g., those emanating from system design requirements).

Using Multi-Object Tracking (MOT) as a case study, we demonstrate how to express domain knowledge, abductively infer domain-specific events and define interpretable ranking criteria through preference-weighted features such as Intersection over Union (IoU) and re-identification (ReID) similarity. Our results confirm that by treating explanation and selection as a competition between plausible models, and guided by preference structures, one can achieve strong empirical performance while offering gains in transparency, modularity, elaboration tolerance, domain adaptation, and flexibility compared to conventional end-to-end pipelines. In other words, we particularly emphasize that our work is not about development of object tracking per se, i.e., neurosymbolically grounded explainable inference may be performed about any high-level visual or spatial feature of interest. Naturally, challenges for this neurosymbolic framework remain. The performance is inherently bounded by the quality of the sub-symbolic perception modules, and while the reasoning is efficient for complex real-world scenarios as demonstrated in our evaluation, the computational overhead from grounding and solving may limit practical scalability in extreme edge cases with a vast number of interacting objects.

Looking outward, opportunities for future work include extending the framework to more complex visual tasks such as event recognition and intent inference, where the interplay between visual perception and symbolic reasoning can be further explored. Scaling to real-time inference, incorporating richer event ontologies, and enabling tighter integration with neural perceptual modules are key steps toward broader applicability. Ultimately, this work contributes to bridging the gap between symbolic reasoning and modern machine perception, offering a declarative and interpretable path forward for dynamic visual understanding.

Dissemination. Relevant materials pertaining to the developed framework (e.g., code, data) as well as future extensions may be obtained via: Cognitive Vision. codesign-lab.org/cognitive-vision.

Acknowledgments. We acknowledge funding by the Swedish Research Council (Vetenskapsrådet – VR) through the project *Counterfactual Commonsense* (Project ID: 2022-02960_VR), as well as funding from the Swedish Foundation for Strategic Research (Stiftelsen för Strategisk Forskning – SSF).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Bartholomew, M., Lee, J.: Functional stable model semantics and answer set programming modulo theories. In: IJCAI 2013, Procee. of 23rd Intern. Joint Conf. on Artificial Intelligence. pp. 718–724. IJCAI/AAAI (2013)
2. Bewley, A., Ge, Z., Ott, L., Ramos, F., Upcroft, B.: Simple online and realtime tracking. In: 2016 IEEE International Conference on Image Processing (ICIP). IEEE (Sep 2016). <https://doi.org/10.1109/icip.2016.7533003>
3. Bhatt, M.: Reasoning about space, actions and change: A paradigm for applications of spatial reasoning. In: Qualitative Spatial Representation and Reasoning: Trends and Future Directions. IGI Global, USA (2012)
4. Bhatt, M., Suchan, J.: Cognitive Vision and Perception: Deep Semantics Integrating AI and Vision for Reasoning about Space, Motion, and Interaction. In: Proceedings of the 24th European Conference on Artificial Intelligence (ECAI 2020), Including PAIS 2020. Frontiers in Artificial Intelligence and Applications, vol. 325, pp. 2881–2882. IOS Press (2020). <https://doi.org/10.3233/FAIA200434>
5. Bhatt, M., Suchan, J.: Artificial Visual Intelligence: Perceptual Commonsense for Human-Centred Cognitive Technologies, pp. 216–242. Springer (2023). https://doi.org/10.1007/978-3-031-24349-3_12
6. Brewka, G., Eiter, T., Truszczyński, M.: Answer set programming at a glance. *Comm. ACM* **54**(12), 92–103 (2011). <https://doi.org/10.1145/2043174.2043195>
7. Gebser, M., Kaminski, R., Kaufmann, B., Schaub, T.: Multi-shot ASP solving with clingo. *Theory and Practice of Logic Programming* **19**(1), 27–82 (2019). <https://doi.org/10.1017/S1471068418000054>
8. Hahn, S., Janhunnen, T., Kaminski, R., Romero, J., Rühling, N., Schaub, T.: Plingo: A system for probabilistic reasoning in clingo based on LP^{mln}. In: Rules and Reasoning - 6th International Joint Conference on Rules and Reasoning, RuleML+RR 2022, Berlin, Germany, September 26-28, 2022, Proceedings. *Lecture Notes in Computer Science*, vol. 13752, pp. 54–62. Springer (2022). https://doi.org/10.1007/978-3-031-21541-4_4
9. Hazarika, S.M.: Qualitative spatial change: space-time histories and continuity (January 2005), <https://etheses.whiterose.ac.uk/id/eprint/1325/>, unpublished. Supplied directly by the School of Computing, University of Leeds.
10. Kowalski, R., Sergot, M.: A Logic-Based Calculus of Events, pp. 23–55. Springer Berlin Heidelberg, Berlin, Heidelberg (1989). https://doi.org/10.1007/978-3-642-83397-7_2
11. Kuhn, H.W.: The hungarian method for the assignment problem. *Naval Research Logistics Quarterly* **2**(1-2), 83–97 (1955). <https://doi.org/10.1002/nav.3800020109>
12. Law, M., Russo, A., Broda, K.: The complexity and generality of learning answer set programs. *Artificial Intelligence* **259**, 110–146 (2018). <https://doi.org/10.1016/j.artint.2018.03.005>
13. Lee, J., Meng, Y.: Answer set programming modulo theories and reasoning about continuous changes. In: IJCAI 2013, Procee. of 23rd Intern. Joint Conf. on Artificial Intelligence. pp. 990–996. IJCAI/AAAI (2013), <http://www.aaai.org/ocs/index.php/IJCAI/IJCAI13/paper/view/6895>
14. Lee, J., Talsania, S., Wang, Y.: Computing LPMLN using ASP and MLN solvers. *Theory Pract. Log. Program.* **17**(5-6), 942–960 (2017). <https://doi.org/10.1017/S1471068417000400>
15. Lee, J., Wang, Y.: Weight learning in a probabilistic extension of answer set programs. In: Principles of Knowledge Representation and Reasoning: 16th Internl.

- Conf, KR 2018. pp. 22–31. AAAI Press (2018), <https://aaai.org/ocs/index.php/KR/KR18/paper/view/18057>
16. Lifschitz, V.: Achievements in answer set programming. *Theory Pract. Log. Program.* **17**(5-6), 961–973 (2017). <https://doi.org/10.1017/S1471068417000345>
 17. Luiten, J., Osep, A., Dendorfer, P., Torr, P., Geiger, A., Leal-Taixé, L., Leibe, B.: Hota: A higher order metric for evaluating multi-object tracking (2020)
 18. Ma, J., Miller, R., Morgenstern, L., Patkos, T.: An epistemic event calculus for asp-based reasoning about knowledge of the past, present and future. In: LPAR-19. 19th International Conference on Logic for Programming, Artificial Intelligence and Reasoning. EPiC Series in Computing, vol. 26, pp. 75–87. EasyChair (2014). <https://doi.org/10.29007/zswj>
 19. Mani, I., Pustejovsky, J.: Interpreting Motion - Grounded Representations for Spatial Language, *Explorations in language and space*, vol. 5. Oxford University Press (2012), <http://ukcatalogue.oup.com/product/9780199601240.do>
 20. Marr, D.: Vision: A Computational Investigation into the Human Representation and Processing of Visual Information. Henry Holt and Co., Inc., New York, NY, USA (1982)
 21. Milan, A., Leal-Taixé, L., Reid, I., Roth, S., Schindler, K.: MOT16: A benchmark for multi-object tracking (Mar 2016), <http://arxiv.org/abs/1603.00831>
 22. Schultz, C., Bhatt, M., Suchan, J., Walega, P.A.: Answer set programming modulo ‘space-time’. In: Rules and Reasoning - Second International Joint Conference, RuleML+RR 2018, Luxembourg, September 18-21, 2018, Proceedings. Lecture Notes in Computer Science, vol. 11092, pp. 318–326. Springer (2018). https://doi.org/10.1007/978-3-319-99906-7_24
 23. Suchan, J., Bhatt, M., Monsen, J.: ASP-Driven Visual Commonsense: A General Framework for Reasoning about Embodied Interaction in the Wild. In: Principles of Knowledge Representation and Reasoning: Proceedings of the 22nd International Conference, KR 2025, Melbourne, Australia, Nov 2025. AAAI Press (2025)
 24. Suchan, J., Bhatt, M., Varadarajan, S.: Out of sight but not out of mind: An answer set programming based online abduction framework for visual sensemaking in autonomous driving. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019. pp. 1879–1885. ijcai.org (2019). <https://doi.org/10.24963/ijcai.2019/260>
 25. Suchan, J., Bhatt, M., Varadarajan, S.: Commonsense visual sensemaking for autonomous driving - on generalised neurosymbolic online abduction integrating vision and semantics. *Artificial Intelligence* **299**, 103522 (2021). <https://doi.org/10.1016/j.artint.2021.103522>
 26. Walega, P.A., Bhatt, M., Schultz, C.P.L.: ASPMT(QS): non-monotonic spatial reasoning with answer set programming modulo theories. In: Logic Programming and Nonmonotonic Reasoning - 13th Internl. Conf., LPNMR 2015., Proceedings. LNCS, vol. 9345, pp. 488–501. Springer (2015). https://doi.org/10.1007/978-3-319-23264-5_41
 27. Wojke, N., Bewley, A., Paulus, D.: Simple online and realtime tracking with a deep association metric. In: 2017 IEEE International Conference on Image Processing (ICIP). pp. 3645–3649. IEEE (2017). <https://doi.org/10.1109/ICIP.2017.8296962>
 28. Yang, Z., Ishay, A., Lee, J.: Neurasp: Embracing neural networks into answer set programming. In: 29th International Joint Conference on Artificial Intelligence, IJCAI-20. pp. 1755–1762. International Joint Conferences on Artificial Intelligence Organization (7 2020). <https://doi.org/10.24963/ijcai.2020/243>