ESRA: A Neuro-Symbolic Relation Transformer for Autonomous Driving

ESRA: a Neuro-Symbolic Relation Transformer for Autonomous Driving

Alessandro Sebastian Russo[§]
DAUIN
Politecnico di Torino
Turin, Italy
alessandro.russo@polito.it

Lia Morra
DAUIN
Politecnico di Torino
Turin, Italy
lia.morra@polito.it

Fabrizio Lamberti
DAUIN
Politecnico di Torino
Turin, Italy
fabrizio.lamberti@polito.it

Paolo Emmanuel Ilario Dimasi[§]
DAUIN
Politecnico di Torino
Turin, Italy
paolo.dimasi@studenti.polito.it

***Abstract*—Scene Graph Generation (SGG) is a powerful tool for autonomous vehicles to understand their environment. In this paper, a novel one-stage neuro-symbolic architecture called nEuro-Symbolic Relation trAnsformer (ESRA) is proposed and its applications to SGG in the field of autonomous driving are investigated. This one-stage architecture can perform both object and relation recognition in a single step, attempting to incorporate prior knowledge in the form of logical propositions grounded by a Logic Tensor Network (LTN). To the best of our knowledge, this is the first attempt to combine a transformer-based architecture with an LTN for SGG. The results show that the integration of LTN increases mean recall (mR) by up to 21% in the best configuration, with mAP achieving an increase of up to 19%.**

***Index Terms*—Autonomous Driving, Logic Tensor Networks, Neuro-symbolic AI, Scene Graph Generation, One Stage Detection, Visual Relationship Detection**

I. Introduction

Autonomous vehicles require a comprehensive understanding of their environment. Therefore, enormous efforts have been made to improve perception tasks such as object detection [1], semantic segmentation [2] or motion prediction [3]. With the aim of achieving comprehensive and holistic scene understanding, scene graph generation (SGG) can prove useful in this scenario. This task goes beyond object detection to encompass all visual relationships between actors and objects in a structured representation [4], [5]. A scene graph, as shown in Fig. 1, is a directed graph that encodes objects and actors as nodes, while edges represent pairwise relationships (e.g. spatial relationships, actions) between them. However, despite the wealth of information that it can provide an autonomous agent, few studies have addressed SGG in the field of autonomous driving [6]–[10], and only recently have appropriately annotated datasets been proposed, such as Traffic Genome [11].

Among other issues, the SGG task suffers from the often incomplete, biased, and long-tailed distribution of available annotations [4], [12], [13]. For a given image, there are several valid scene graphs, and the available annotations usually cover only a subset of all possible solutions. Furthermore, the distribution of relationship labels is often skewed, e.g., classes such as 'standing on' or 'right of' appear more frequently than

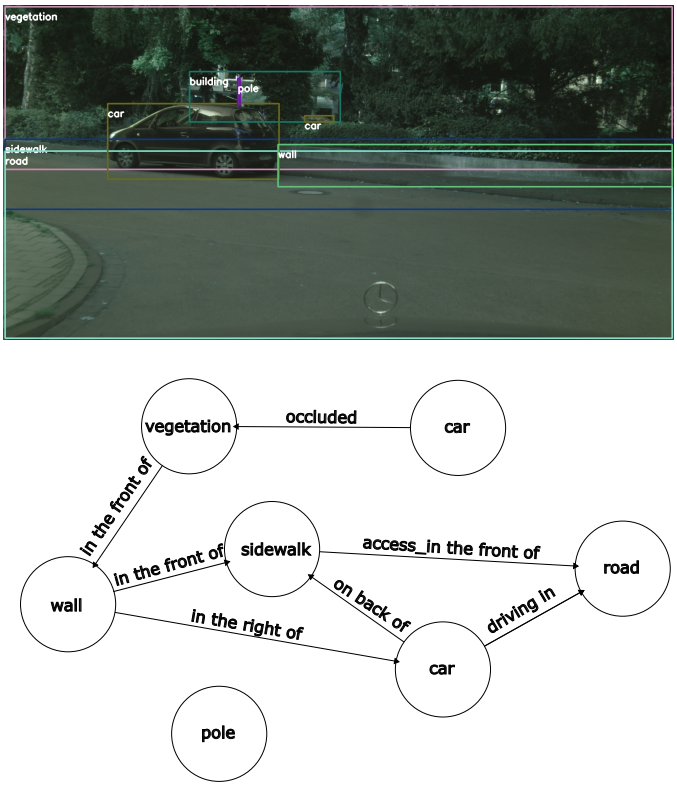


Fig. 1: Example of a scene graph (bottom) extracted from an image (top), with bounding boxes of the detected objects super-imposed

classes such as 'hanging from' or 'back right of'. This is in part due to their inherent long-tail distribution [4] and in part due to the natural preference of annotators for more general - and thus less informative - labels [12]. In the literature, many techniques have been proposed to compensate for the bias of SGG training sets. Among these, neuro-symbolic (NeSy) techniques have shown promising results [13]. NeSy is a subfield of artificial intelligence that aims to integrate and incorporate concepts and methods of knowledge representation, reasoning, and statistical machine learning to develop robust and scalable learning systems. In particular, NeSy frameworks such as Logic Tensor Networks [14] can compensate for incomplete or

[§]These authors contributed equally to this work

imperfect label supervision by incorporating prior knowledge and constraints in the form of first-order language (FOL statements) into the learning process (e.g., the information that humans can drive vehicles and not vice versa, or the notion that certain classes are subcategories of more general classes) [13], [15]. However, previous work has mainly focused on two-stage methods, where the recognition of objects and their relationships is approached by two disjoint networks [13]. More recently, one-stage methods have emerged with the goal of solving both tasks (object detection and visual relationship detection) simultaneously. One-stage models are attractive due to their faster inference time, lower computational cost, increased simplicity, and potentially higher overall accuracy [5], [7].

This paper proposes a novel one-level neuro-symbolic architecture called **nEuro-Symbolic Relation trAnsformer (ESRA)** and explores its applications to SGG in the field of autonomous driving. This paper extends previous NeSy experiments [13] in two ways: (i) it introduces a one-stage architecture that computes both object and relationshiprecognition in a single step, aiming to take into account the prior knowledge available in the form of logical constraints, and (ii) it performs experiments on larger datasets (Traffic Genome and Visual Genome) tailored to the autonomous driving domain.

The rest of the paper is organized as follows. Section II places this work in the context of related literature and gives a background on LTNs. Section III describes in detail how ESRA combines LTNs with a state-of-the-art transformer-based one-stage SGG architecture. Subsequently, sections IV and V examine the behavior of the model on Traffic Genome under different configurations. Finally, conclusions and future work are discussed in section VI.

II. Background and related work

A. Logic Tensor Networks

Logic Tensor Networks (LTNs), first introduced by Serafini and Garcez [16], integrate FOL into a differentiable framework based on real logic to allow the embedding of FOL instructions into deep learning architectures. Real Logic is based on a first order language $\mathcal{L}$ whose signature consists of a set $\mathcal{C}$ of constant symbols, a set $\mathcal{F}$ of functional symbols, a set $\mathcal{P}$ of predicate symbols and a set $\mathcal{X}$ of variable symbols. Real Logic uses the sets of $\mathcal{L}$ to express relational knowledge based on fuzzy connectors and it grounds them as tuples of real numbers.

Formally, a *grounding* $\mathcal{G}$ for a first order language $\mathcal{L}$ is a function from the signature of $\mathcal{L}$ to the space of real numbers that satisfies the following conditions:

1. $\mathcal{G}(c) \in \mathbb{R}^n$ for every constant symbol $c \in \mathcal{C}$.
2. $\mathcal{G}(f) \in \mathbb{R}^{n \cdot m} \longrightarrow \mathbb{R}^n$ for every function $f \in \mathcal{F}$ where m is the arity of f.
3. $\mathcal{G}(P) \in \mathbb{R}^{n \cdot m} \longrightarrow [0,1]$ for every predicate $P \in \mathcal{P}$ where m is the arity of P.

Formulas are defined by the combination of connectors, predicates and quantifiers. The collection of closed formulas, including axioms, forms a knowledge base denoted as $\mathcal{K}$. The learning problem in LTNs is formulated as a best satisfiability problem, where the goal is, given a grounding $\hat{\mathcal{G}}_\theta$ (where θ represents the parameters of all predicates), to identify the values of Θ^* that maximize the truth values of the conjunction of all closed formulas $\phi \; in \mathcal{K}$.

$$\Theta^* = \text{argmax}_\Theta \, \hat{\mathcal{G}}_\theta \left(\bigwedge_{\phi \in \mathcal{K}} \phi \right) - \lambda \|\Theta\|_2^2 \tag{1}$$

LTNs have been integrated into deep learning architectures with several objectives. A common approach is to use them to redefine the learning task, as LTNs maximize the satisfiability of logical statements that can represent both known facts (what would be denoted as labeled examples in a standard machine learning framework) and axioms that most predictions should satisfy (what is generally considered prior knowledge, commonsense knowledge, or domain-specific constraints) [13], [15], [17]. Other works have instead integrated an LTN to incorporate prior knowledge at inference time, refining model predictions [18]. The present work falls under the former category.

B. Scene Graph Generation

Given a set of object classes $\mathcal{C}$, a set of attribute types $\mathcal{A}$, and a set of relationship types $\mathcal{R}$, a scene graph $\mathcal{H}$ is a tuple $\mathcal{H} = (\mathcal{O}, \mathcal{E})$ where $\mathcal{O} = \{o_1, ..., o_n\}$ is a set of objects and $\mathcal{E} \subseteq \mathcal{O} \times \mathcal{R} \times \mathcal{O}$ is a set of edges. Each object has the form $o_i = (c_i, A_i)$ where $c_i \subseteq \mathcal{C}$ is the class of the object and $A_i \subseteq \mathcal{A}$ are the attributes of the object, although attributes are often omitted when modelling the SGG task. Note that the graph is directed since relationships are generally not symmetric: a car can drive on the road, but the contrary is impossible.

SGG architectures can be clustered into two main classes. While earlier works featured predominantly a two-stage approaches, in which the object and relationship detection modules are trained separately, more recent works have started shifting to an end-to-end one-stage approaches, in which the object detection and relationship prediction tasks are jointly modeled and trained [4], [5].

As mentioned in Section I, several issues in visual relationship detection datasets negatively affect training:

a) Missing and noisy annotations: Both Visual Genome Dataset [19], the most commonly used SGG benchmark, and Traffic Genome [11] have missing annotations, i.e., the annotations typically contain only one predicate for a pair of objects, even if other choices would be equally valid. In fact, it would be practically impossible to collect all possible annotations for a given image. This restriction means that correct triplets that do not appear in the dataset during training are rejected.

b) Long-tailed relationship distribution: the relationship labels usually follow an exponential distribution. As a result, most SGG models end up predicting the most frequently occurring classes and disregard the remaining ones.

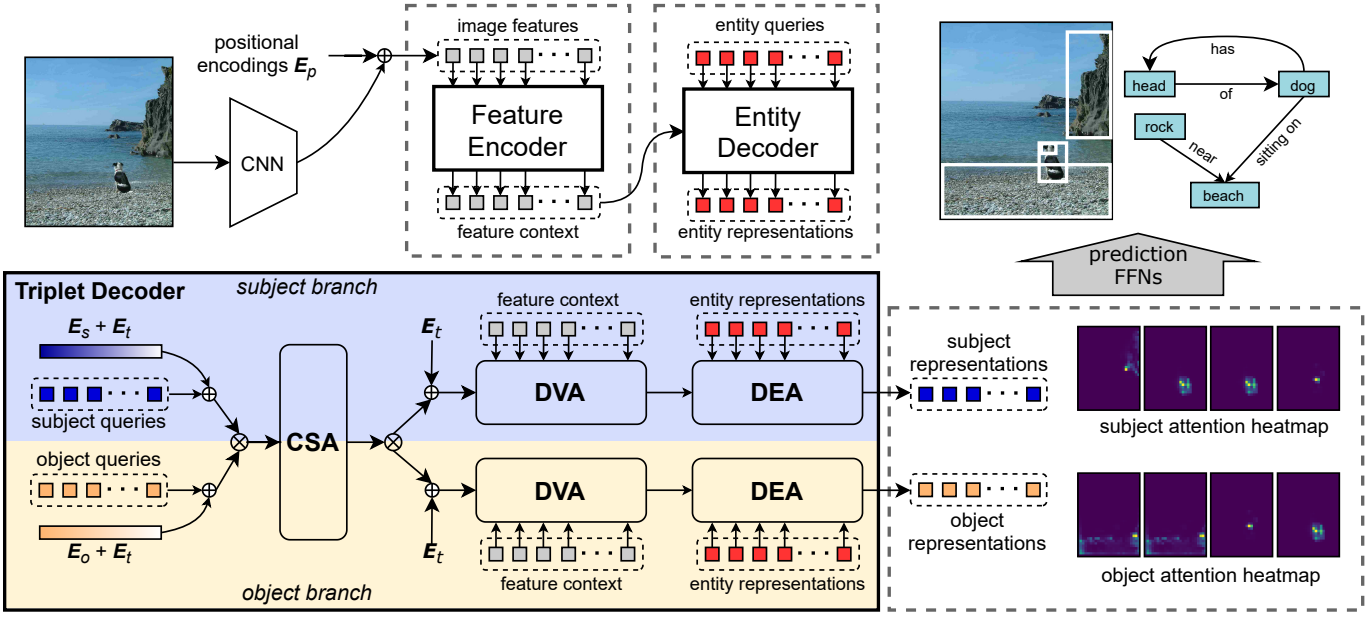


Fig. 2: Schema of the RelTR architecture reproduced from [5]. Given a set of learned subject and object queries coupled by subject and object encodings, RelTR captures the dependencies between relationships and contextual information. Feature context and entity representation, encoded respectively by the output of the feature encoder and entity decoder, are used to directly compute a set of subject and object representations. A pair of subject and object representations with attention heat maps is decoded into a triplet ⟨subject, relationship, object⟩ by feedforward networks (FFNs). CSA, DVA and DEA stand for Coupled Self-Attention, Decoupled Visual Attention, and Decoupled Entity Attention. E_p, E_t, E_s and E_o are the positional, triplet, subject and object encodings, respectively. $\oplus$ indicates element-wise addition, while $\otimes$ indicates concatenation or split.

Various techniques have been introduced to mitigate these problems. Statistical learning-based approaches counteract bias in the dataset distribution by exploiting techniques such as Conditional Random Field [20] and Markov Logic Networks [21]. **Integrating External Knowledge** from other sources, such as knowledge graphs and linguistic priors, can reduce the effect of bias through techniques such as Knowledge Graph Embedding [22], [23], Distillation Knowledge [24], Visual-linguist Prior [25], [26], or Neuro-symbolic integration [13]. **Novel Losses** were proposed, tailored to SGG, based on contrastive losses [27] or causal reasoning [12]. **Tailored Message Parsing** and Graph Neural Network were also employed to enhance relationship prediction [28].

This paper aims to show how a NeSy framework, such as LTNs, can be used in conjunction with a one-stage end-to-end framework to introduce prior knowledge in the form of FOL statements during the training process. The LTN balances the evidence from the ground-truth annotation with such constraints to facilitate training.

C. SGG in Autonomous Driving

In the field of autonomous driving, existing approaches to SGG face remarkable challenges. Previous studies often focus only on specific subtasks, such as topology reasoning for lanes and traffic elements [6], [7] neglecting key aspects such as pedestrian interactions. Others focus on risk assessment for specific actions in traffic scenarios [8], [9], lacking sufficient generalization. Some approaches define manual rules to extract spatial relationships [6], [10], but have difficulties to generalize across different traffic scenarios. Due to the specific nature of these techniques, they cannot be easily applied to other datasets or compared or combined with state-of-the-art techniques trained on non-autonomous driving datasets. Zhang et al. introduce a public benchmark dataset, Traffic Genome, which provides a comprehensive traffic scenario with various entities and relationships [11], paving the way for a more sophisticated investigation of the SGG task in autonomous driving. However, they only evaluated the performance of two-stage methods, which are known for their complexity and low speed.

III. Methodology

In this section the proposed nEuro-Symbolic Relation trAnsformer (ESRA) is presented. ESRA is based on the Relation Transformer Network (RelTR) [5] due to its state-of-the-art result in one-stage SGG. ESRA combines RelTR loss with a NeSy component as follows:

$$\mathcal{L}_{\text{ESRA}} = \mathcal{L}_{\text{RelTR}} + \alpha \mathcal{L}_{\text{NeSy}} \tag{2}$$

where α is an hyperparameter that controls the influence of the NeSy component $\mathcal{L}_{\text{NeSy}}$ during training. The rest of the section is organized as follows: Section III-A briefly reviews the RelTR architecture; Section III-B introduces the knowledge base used in ESRA, while Section III-C presents different policies for computing its truthiness and Section III-B1 illustrates the chosen grounding.

A. Relation Transformer Network

RelTR [5], illustrated in Figure 2, uses an encoder-decoder architecture based on the DEtection TRasformer or DETR [29], where the encoder reasons about the visual feature context and the decoder infers a fixed-size set of ⟨ subject, object, relationship ⟩ triplets using different types of attention mechanisms. The key aspects are summarized here, and the reader is referred to the original paper for further details [5].

The loss function of RelTR is formulated as follows:

$$\mathcal{L}_{\text{RelTR}} = \mathcal{L}_{\text{sub}} + \mathcal{L}_{\text{obj}} + \mathcal{L}_{\text{cls}}^{\text{prd}} \tag{3}$$

$$\mathcal{L}_{\text{sub}} = \sum_{i=1}^{N_t} \Theta \left[\mathcal{L}_{\text{cls}} + 1_{\left\{c_{\text{sub}}^i \neq \phi\right\}} \mathcal{L}_{\text{box}} \right] \tag{4}$$

$$\mathcal{L}_{\text{obj}} = \sum_{i=1}^{N_t} \Theta \left[\mathcal{L}_{\text{cls}} + 1_{\left\{c_{\text{obj}}^i \neq \phi\right\}} \mathcal{L}_{\text{box}} \right] \tag{5}$$

where $\mathcal{L}_{\text{cls}}^{\text{prd}}$ is the cross-entropy loss for predicate classification, $\mathcal{L}_{\text{sub}}$ the cross-entropy loss for subject classification, $\mathcal{L}_{\text{cls}}$ the cross-entropy loss for object classification, c_{sub}^i the i-th class object and c_{sub}^i the i-th class subject. Θ is 0 when *background* class is assigned to a poor object prediction, i.e. the box overlaps with the ground-truth bounding box IoU above a predefined threshold of another entity label. In the other cases, Θ is always 1. The success of RelTR is based on 3 modules:

1) Coupled Self-Attention (CSA): captures the context between triplet proposals and the dependencies between subjects and objects in SGG using their latent encoding E_t, E_s, E_o, which are learned during training.

2) Decoupled Visual Attention (DVA): extracts visual features independently for subject and object query representation. DVA operates in a decoupled manner, where the computations of subject and object representations are independent of each other. This approach allows for the extraction of fine-grained visual information and enhances the localization and classification of subjects and objects.

3) Decoupled Entity Attention (DEA): improves the localization and classification of subjects and objects by utilizing entity detection results from the entity decoder. ESRA then feeds the LTN the logits of the feed-forward network to compute the NeSy loss.

B. Knowledge base

Similarly to previous work [13], the developed knowledge base $\mathcal{K}_{ESRA}$ imposes prior knowledge through domain range constraints that list the semantic classes that can (positive range) or cannot (negative) be the subject or object of a relationship.

1) Grounding: The grounding of each term *t* in the triplet ⟨ subject, object, relationship ⟩ within an image I is the logit f_t extracted from the last layer of the RelTR network:

$$\mathcal{G}_{ESRA}(t) = f_t(I) = \hat{t} \tag{6}$$

The grounding of predicates is done as follows:

$$\mathcal{G}(\texttt{IsOfClass}) : x, l \rightarrow l^{\intercal}\texttt{softmax}(x) \tag{7}$$

$$\mathcal{G}(\texttt{InSet}) : x, \mathcal{S} \rightarrow \sum_{s \in \mathcal{S}} s^{\intercal}\texttt{softmax}(s) \tag{8}$$

where x and l represent the predicted class and one-hot encoded label, and $\mathcal{S}$ is a set of labels denoted by the combination of the corresponding one-hot encodings. The grounding of logical connectives is discussed in Section IV.

2) Predicates: Two types of predicates are defined: $\texttt{IsOfClass}(x, l)$ evaluates the class-membership of a given entity x with respect to the label l, while $\texttt{InSet}(x, s)$ computes the set-membership, i.e., whether an entity x belongs to a predefined set of classes s. While the former is used to compute the actual scene graph, the latter facilitates the introduction of domain and range constraints on the relationships.

Let $\langle x, y, z \rangle$ be a network prediction, a triplet with subject x, object y and relationship z, and l_x, l_y, l_z the corresponding ground truth labels. Then, let $\mathcal{S}_t(l_z)$ and $\mathcal{W}_t(l_u)$ be the set of admissible and inadmissible labels, respectively, for each element t in the visual relationship triplet, where t and u are used to denote the subject x, the object y or the relationship z: for example, $\mathcal{S}_x(l_z)$ represents the set of admissible labels for the subject x given a relationship identified by the class label l_z.

3) Axioms: The axioms in $\mathcal{K}_{ESRA}$ were formalized as follows[1]:

$$\phi_1 : \forall \text{Diag}\,(z, l_z)\,(\texttt{IsOfClass}(z, l_z)) \tag{9}$$

$$\begin{aligned} \phi_2 : \forall \text{Diag}\,(\langle x, y, z \rangle, l_z,)& \\ \Big(\texttt{IsOfClass}(z, l_z)& \\ \implies \texttt{InSet}(x, \mathcal{S}_x(l_z)) \wedge \texttt{InSet}(y, \mathcal{S}_y(l_z))\Big)& \end{aligned} \tag{10}$$

$$\begin{aligned} \phi_3 : \forall \text{Diag}\,(\langle x, y, z \rangle, l_x, l_y,)& \\ \Big(\texttt{IsOfClass}(x, l_x) \wedge \texttt{IsOfClass}(y, l_y)& \\ \implies \neg\texttt{InSet}(z, \mathcal{W}_z(l_x, l_y))\Big)& \end{aligned} \tag{11}$$

$$\begin{aligned} \phi_4 : \forall \text{Diag}\,(\langle x, y, z \rangle, l_z, \mathcal{W}_x, \mathcal{W}_y)& \\ \Big(\texttt{IsOfClass}(z, l_z)& \\ \implies \neg\texttt{InSet}(x, \mathcal{W}_x(l_z)) \wedge \neg\texttt{InSet}(y, \mathcal{W}_y(l_z))\Big)& \end{aligned} \tag{12}$$

ϕ_1 verifies the class membership of the predicted relationships with respect to the reference standard. ϕ_2 encodes the *positive range constraints* and is enforced on relationships, while ϕ_3 and ϕ_4 are two alternative methods to encode the *negative range constraints*. For example, suppose that the network predicts a triplet such as $\langle sky, road, watching \rangle$. ϕ_4 is not satisfied because the subject (*sky*) is in the inadmissible set $\mathcal{W}_y(watching)$; ϕ_3 is also not satisfied ($watching$ is not an admissible relationship for the pair formed by the sky subject and $road$ object). Similarly, ϕ_2 is not fulfilled because the set of admissible subjects for the relationship *watching* includes persons but not inanimate objects.

[1]Diagonal Quantification (Diag) quantifies over pairs of instances, e.g., images and their labels. A more formal definition can be found in [14].

TABLE I: Selected set of relationships involved in the range constraints. Relationships belonging to the tailed set are *italicized*

Relationships
driving_in, in_front_of ,in_the_left_of, in_the_right_of, in/on
above, access_around, access_between, access_in_the_center_of, access_in_the_front of, access_in_the_left_of, watching,
access_in_the_right_of, access_in_two_side_of, along, attached_to, behind, belonging_to, between, carrying, with,
growing_on, hanging_from, has, holding, in_the_back_left_of, in_the_back_right_of, in_the_front_left_of, standing_on, to
in_the_front_right_of, near, occluded, occluding, on_back_of, over_something, part_of,ride_on, riding, sitting_on, under

4) Range Constraints Definition: In order to compute the axioms ϕ_2, ϕ_3, and ϕ_4, it is necessary to define the sets of admissible/inadmissible $\langle$ subject-object-relationship $\rangle$ triplets. Two strategies were tested, one completely automatic that relies on existing annotations and one that relies on a manual annotation phase. In the automatic strategy, the admissible and inadmissible sets are generated from the ground-truth annotations, assuming that a particular subject-relationship or object-relationship pair combination occurring in the training set must be admissible, while the opposite holds for inadmissible combinations. Using this strategy, a total of 1,572 admissible and 22,732 inadmissible triplets were determined. In the manual strategy, instead, all possible subject-relationship and all possible relationship-object pairs were reviewed, to determine which would be admissible or inadmissible in a real-world scenario. Then, the set of inadmissible relationships for a given subject-object pair (denoted as $\mathcal{W}_z(l_x, l_y)$ in axiom ϕ_3) are generated by combining the inadmissible subject-relationship ($\mathcal{W}_z(l_s)$) and object-relationship pairs ($\mathcal{W}_z(l_o)$). This resulted in a significant increase in the admissible set, which includes a total of 8,526 admissible and 15,778 inadmissible triplets. Finally, an additional set of admissible/inadmissible triplets was extracted for both annotation methods, containing only relationships that are tailed in the Traffic Genome distribution. For the automatic strategy, this set contains 1,246 admissible and 26,978 inadmissible triplets; for the manual strategy, it contains 12,588 admissible and 15,636 inadmissible triplets.

C. Policy for Domain and Range Constraints

The axioms in $\mathcal{K}_{ESRA}$ are explicitly designed to enforce the consistency in the predictions of the relationship component of each triplet $\langle$ subject, object, relationship $\rangle$ after the network RelTR has computed an initial prediction based on the standard cross-entropy loss. The LTN axioms focus specifically on the relationship class prediction, as object detection is a simpler and less ambiguous task for which performance is generally very high. Nonetheless, there are cases where the prediction of the subject and object class is incorrect, or where the background may lead to incorrect predictions. In this case, it is unclear how to determine the correct range and domain constraints to apply.

Specifically, two policies for calculating the range constraints were compared experimentally: the **Ground Truth policy** evaluates the degree of truthfulness based on the ground truth subject and object labels, while the **Prediction policy** evaluates it based on the predicted subject and object labels. For example, suppose that the predicted triplet is $\langle person, over, car\rangle$, while the corresponding ground truth is $\langle person, over, road\rangle$: in this case, the ground truth policy checks whether *car* is an appropriate object for *over*, while the prediction policy checks the consistency between *over* and *road*. In other words, the first policy emphasizes consistency with ground-truth annotations, while the second focuses on finding coherent subject and object predictions for the relationship. The two policies coincide if the predictions for the object and subject are correct.

TABLE II: Mean Average Precision (mAP) and mean Recall (mR) under different policy and NeSy loss weights on the validation set. Focusing on the coherence of prediction shows better results than focusing on the coherence of ground-truth. The reduction of the NeSy loss weight further emphasizes this result.

Model	α	**Policy**	mAP@0.50	mR@100
ESRA	1	prediction	**21.1**	**26.4**
	1	ground-truth	**21.1**	21.6
	0.1	prediction	**21.0**	**26.6**
	0.1	ground-truth	20.5	23.3
RelTR	-	-	20.3	23.2

Lastly, differently from the original RelTR architecture, non-maximum suppression (NMS) is applied to restrict evaluation only to significantly different bounding boxes.

IV. Experimental Settings

A. Dataset

A detailed analysis of Traffic Genome [11] is shown in Figure 3. Given the small size of the dataset, training transformers can be difficult as they are very data intensive. To overcome this challenge, the Traffic Genome training dataset was augmented with a subset of the Visual Genome dataset [19]. In contrast to Traffic Genome, Visual Genome covers a broader visual domain than Autonomous Driving. Therefore, its integration requires both pruning and taxonomy alignment. Since manual inspection is impractical, a heuristic rule was defined, for which a scene belongs to an autonomous driving scenario if one of the following classes are present:

a) entity: bike, bus, car, motorcycle, sidewalks, truck, vehicle;

b) relationship: above, against, along, at, attached to, behind, belonging to, between, carrying, covered in, growing on, hanging from, has, holding, in, in front of, looking at, lying on, mounted on, near, on, on back of, over, parked on,

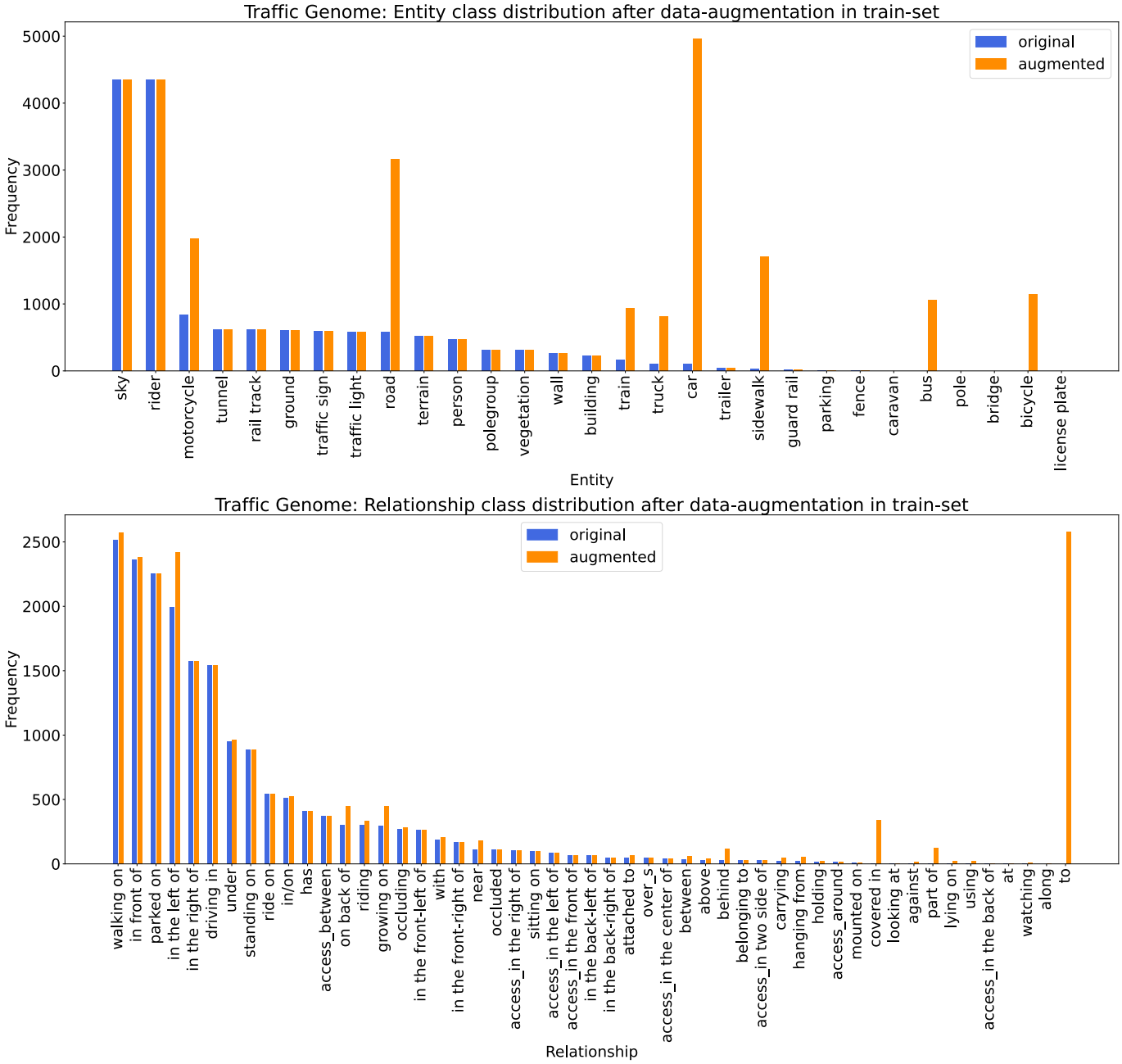


Fig. 3: Comparison of entity (top) and relationship (bottom) frequency before and after extending the Traffic Genome dataset with domain-related images from Visual Genome.

part of, riding, sitting on, standing on, under, using, walking on, watching, with.

If there is no direct match between a Visual Genome and a Traffic Genome label, the taxonomy alignment module replaces the class of the label with the macroclass from Traffic Genome; if this is not possible, the triplet is discarded. Thus, the final training set grew from 630 to 3,017 images and from 18,805 to 23,092 labeled relationships. However, as shown in Figure 3, the persisting tailed distribution of both entities and relationships shows that the expansion of the training set only partially mitigates the biases. For Traffic Genome, the training and test split proposed by the authors was used, reserving 10% of the training set as validation; scenes from Visual Genome were added to the training split.

B. Hyperparameter settings

1) RelTR (baseline) hyper-parameters: The dimension of the input feature map was set to 2048 and the hidden dimension of the entity decoder to 256. The number of layers for the encoder and decoder was set to 5 and 6, respectively. The class cost of the matcher module was set to 1, the bounding boxes to 8, GIoU to 2 and the IoU threshold to 0.7. During

training, the feature encoder layers were kept frozen. The loss weights were set to 5 for the bounding box loss, 2 for the GIOU and 1.5 for the relationship loss, while the weight of the End-Of-Sequence Token was set to 0.4. The maximum number of triplets and bounding boxes was set to 140 for both training and inference to maintain a safe margin between the maximum available data set per image (131 entities and 135 relationships) and the predictable entities. A one-cycle cosine learning rate scheduler was used with a minimum learning rate of 1e-5 and a maximum learning rate of 1e-3 and a warm-up time of 20 epochs.

2) ESRA (LTN) implementation and hyper-parameters: While axiom ϕ_1 was evaluated for all relationship classes, to reduce training time and complexity, ϕ_2, ϕ_3 and ϕ_4 were applied to a broader subset consisting of 41 randomly selected relationships. Additionally, ϕ_3 and ϕ_4 were also applied to another set containing only the tailed relationships from Traffic Genome [11]. The complete list of selected relationships for range constraints is reported in Table I.

a) Grounding of the logical connectives: The fuzzy connectors used were diagonal quantification, stable Reichenbach implication, and Łukasiewicz And and Or, defined in previous work [14]. The generalized p-mean was used for both the aggregator in Eq. 1 ($p = 2$) as well as for the universal quantifier. Experimentally, it was found that scheduling $p_\forall$, starting from $p_\forall = 2$ and doubling it every 50 epochs was the most effective choice for the universal quantifier. In fact, higher values of p increase the contribution of samples that do not satisfy a given axiom, which can lead to premature overfitting in the early stages of training, but place more weight on incorrectly predicted samples as training progresses. α was set to 0.1, while the NMS threshold was set to 0.5 to discard similar bounding boxes.

b) Relationship stratification: Moving from the premise that not all relationships are equally difficult to learn, and in an effort to reduce the computational complexity of the constraints, we examined several variants for the ϕ_3 and ϕ_4 axioms. In the first one, denoted as **Difficulty Stratification**, the value of the $p_\forall$ hyper-parameter was doubled for harder-to-learn relationships, compared to the others (e.g., two different schedules are used $p_{\forall,hard} = [4, 8, ..]$ and $p_{\forall,easy} = [2, 4, ..]$). Hard classes were defined based on the imbalance (i.e., classes with less than 100 examples were denoted as hard). In the second variant, denoted **Set Restriction**, only the constraint set obtained from the tailed relationships in the Traffic Genome dataset described at the end of Section III-B is used, under the assumption that the most frequent relationships are represented by enough examples in the dataset so that the network does not need additional logical constraints to learn them.

3) Hardware: Training has been performed on an NVIDIA(c) RTX 4090 with 24GB of VRAM for 170 epochs with a batch size of 10. The PyTorch version used was 2.1.0, with CUDA version 11.8. LTN experiments were implemented using the LTNtorch library[2].

[2]available from https://github.com/tommasocarraro/LTNtorch

TABLE III: Comparison of the performance with different constraint configuration strategies on the validation set. Configurations including the ϕ_3 axiom obtained higher performances compared to those including the ϕ_4 version.

Negative type	Strategy	mAP@0.50	mR@100
ϕ_3 constraint	base	21.0	26.6
	difficulty stratification	21.3	**29.1**
	set restriction	**22.5**	28.3
ϕ_4 constraint	base	20.8	22.4
	difficulty stratification	**21.3**	21.7
	set restriction	20.4	**23.8**
baseline (RelTR)	-	20.3	23.2

C. Evaluation

mean Average Precision (mAP) and mean Recall (mR) were used for SGG [5]. Note that mR is evaluated according to entities' Intersection over Union (IoU) with a threshold of 0.5. Results were also compared with the Predicate CLaSsification (PredCLS) task of the original RelTR paper for the final baseline comparison in Section V-D.

V. Results

In this section, the results are presented by first comparing the two policies using automatically generated constraints with the negative axiom ϕ_3, followed by a comparison between the negative axioms ϕ_3 and ϕ_4 for both normal and tailed relationships, and finally a comparison between automated and manual constraints. Finally, the best ESRA configuration is compared on the Traffic Genome testing set with the RelTR baseline trained on the augmented Traffic Genome dataset using the hyper-parameters described in Section IV-B1, with additional qualitative results.

A. Comparison of different policies for domain and range constraints

Experimental results for the ground-truth and prediction policies (defined in Section III-C) are presented in Table II. Only the negative axiom ϕ_3 and automatic range constraints were used in this setting. We found experimentally that emphasizing the coherence with the ground-truth subject and object classes (mR@100 = 21.6) as opposed to the predicted classes (mR @ 100 = 26.4) has a detrimental effect on the learning process. In addition, different hyper-parameter configurations lead to similar values for mAP but different mR values. This difference may arise due to the NeSy formulation: in fact, only one axiom is enforced on the entities predictions, while several axioms are enforced on the relationship predictions. As a result, ESRA may prioritize relationship detection over entity detection to achieve an optimal solution. This hypothesis is supported by the validation losses, observing that the cross-entropy loss of entities decreases (0.3 - 0.2), while the relationship loss increases (1.4 - 2.1) in the last 20 epochs of training.

B. Negative Axioms Analysis

The knowledge base defined in Section III may contain redundant constraints, in particular ϕ_3 and ϕ_4. The present

TABLE IV: Relationship recall score comparison between RelTR and ESRA in its ϕ_3 Set Restriction configuration. Due to space constraints, only a sample set of relationships is given. The highlighted scores display how ESRA gained improved performances on most relationships.

Relationship	$R_{val,RelTR}$@100	$R_{val,ESRA}$@100
access_between	90.2	**91.3**
access_in the front of	27.0	**37.0**
access_in the left of	25.0	**37.5**
access_in the right of	**46.2**	**46.2**
access_in two side of	0	**100.0**
driving in	69.1	**75.4**
growing on	**64.5**	61.8
has	50.0	**51.7**
in front of	52.8	52.5
in the front-left of	**7.7**	**7.7**
in/on	60.0	**61.0**
near	**40.0**	20.0
on back of	11.1	**22.2**
over_something	0	**37.5**
ride on	41.7	**50.0**
standing on	59.1	**64.4**
under	35.6	**44.1**
walking on	79.8	**85.5**
with	**58.3**	31.7

analysis investigates which version, ϕ_3 or ϕ_4, yields the highest performance. Again, only automatic annotations were used in this analysis. Here, to determine the applicability of ϕ_3 and ϕ_4 axioms in different configurations, the two additional strategies described in Section IV-B2b were also used.

Table III shows a comparison between ϕ_3 and ϕ_4 with their respective configurations and the RelTR baseline. Results show that the base ϕ_3 configuration is superior to the the base ϕ_4 one. Another interesting observation can be made by looking at the results of the *Difficulty Stratification* and *Set Restriction* configurations: in general, the harder configurations lead to better results than the easier ones. Furthermore, for the Set Restriction configuration, it is interesting to note how imposing constraints only on tailed relationships leads to better results in both versions. In particular, by forcing the network to focus exclusively on challenging tailed examples, the predictions for different relationships seem to become more balanced, as can also be seen in Table IV, where the recall values for a subset of the relationships of RelTR and ESRA were compared.

C. Annotation Method Analysis

Finally, the manual annotations were assessed to determine how much the additional manual work might affect performance. The results are shown in Table VI. Both ϕ_3 and ϕ_4 negative axioms were tested. The results show that the additional manual work significantly increases performance for both negative axiom types, with ϕ_3 achieving the largest increase at mAP (from 21 to 24.3), while ϕ_4 achieves up to 20% higher performance at mR@100 (from 22.4 to 28.2). Moreover, the performances of the manual annotations show another important finding: the knowledge provided by the original dataset labels is a major limitation for the network prediction capabilities, which could be overcome here thanks to the additional knowledge integration provided by a NeSy approach such as the LTN framework.

TABLE V: Performance comparison between the two annotation methods under different strategies of negative constraint on the validation set. Both ϕ_3 and ϕ_4 benefit greatly from the additional manual work, with the latter showing the greatest improvement in mean Recall (mR).

Negative type	Annotations	mAP@0.50	mR@100
ϕ_3 constraint	automatic	21.0	26.6
	manual	**24.3**	**27.2**
ϕ_4 constraint	automatic	20.8	22.4
	manual	**23.0**	**28.2**
baseline (RelTR)	-	20.3	23.2

TABLE VI: Performance comparison between the two annotation methods under different strategies of negative constraint on the validation set. Both ϕ_3 and ϕ_4 benefit greatly from the additional manual work, with the latter showing the greatest improvement in mean Recall (mR).

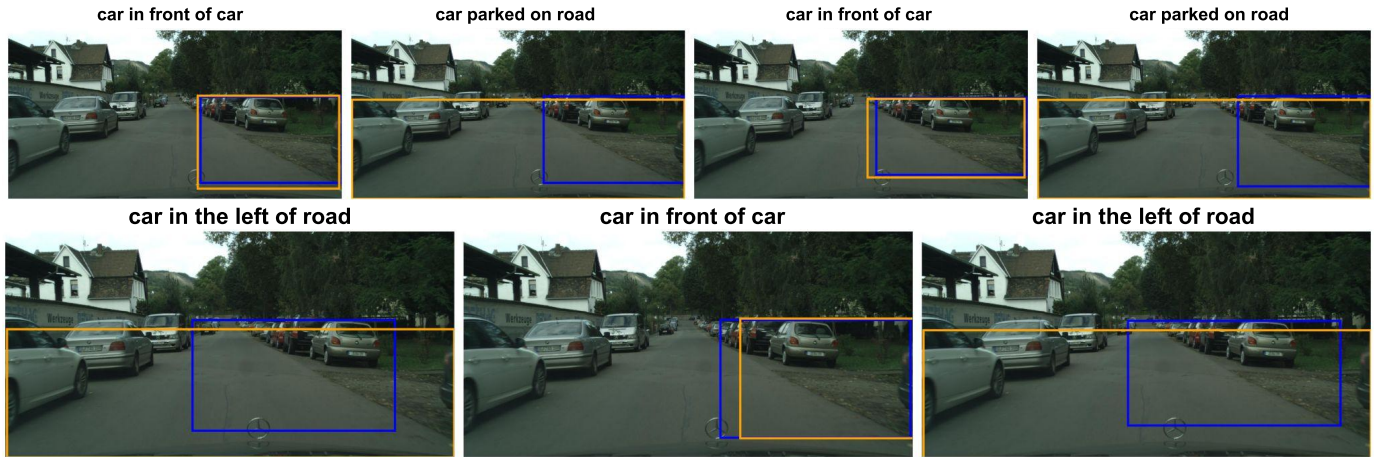
Model	mR@20	mR@50	mR@100
ESRA	**11.3**	**17.8**	**22.8**
RelTR	**11.7**	16.6	18.8

D. Final Result Comparison

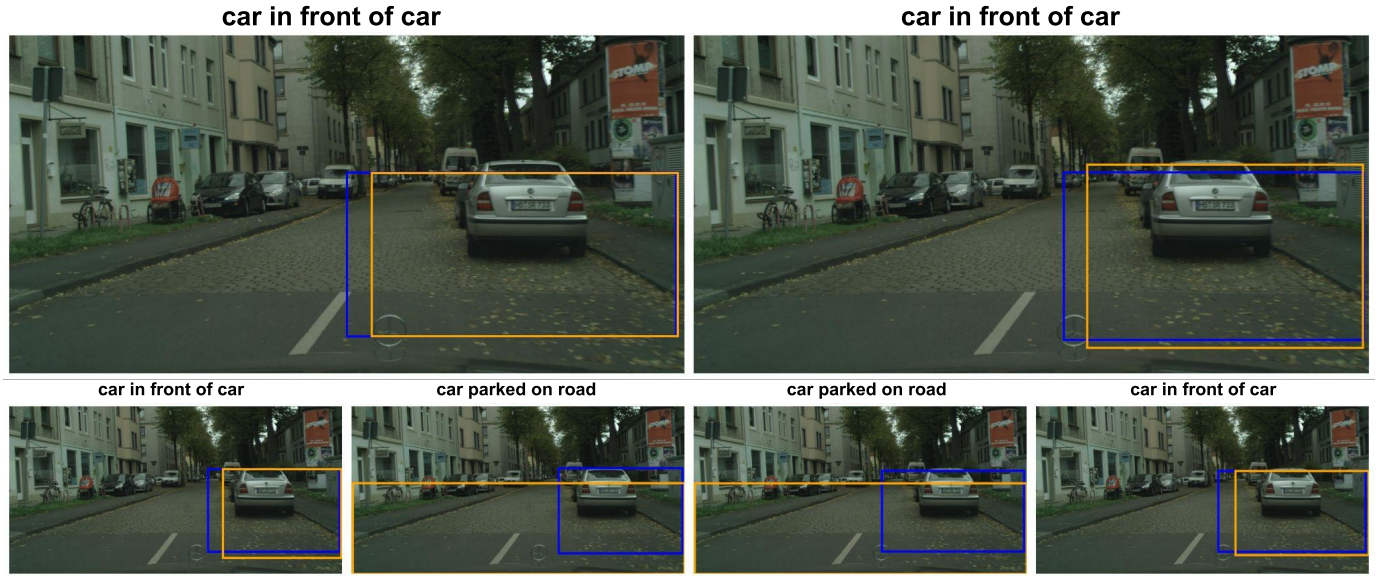
The best ESRA configuration, using the ϕ_3 negative axiom and manually-defined constrained, was compared to the RelTR baseline on the Traffic Genome test set. For consistency with the Traffic Genome paper [11], the evaluation was performed using the PredCLS task recall metric, which refers to relationship detection; an insight into the difference between these metrics can be found in [30]. ESRA outperform RelTR at mR@50 (**ESRA = 17.8, RelTR = 16.6**) and mR100 (**ESRA = 22.8, RelTR = 18.8**), with comparable performances at mR@20 (**ESRA = 11.3, RelTR = 11.7**). One possible explanation is that transformers need comparably larger training sets than convolutional architectures. Adding constraints through a NeSy component may mitigate this phenomenon, preventing the network from attending to irrelevant or incorrect features when predicting relationships.

An additional qualitative comparison between ESRA and the RelTR baseline is shown in Figure 4. The confidence score threshold was set to $p >= 0.8$. Figures 4a and 4b illustrate that both methods work well for classes that appear frequency in the dataset. It can be observed that the baseline sometimes outputs duplicate triplets, a behavior already observed by the authors of RelTR [5]. It is possible that the transformer module does not satisfy the constraint that subject and object cannot be the same entity, and the post-processing module is not capable of consistently solving this issue. This issue could be fixed by introducing in the knowledge base an additional logical constraint in future works.

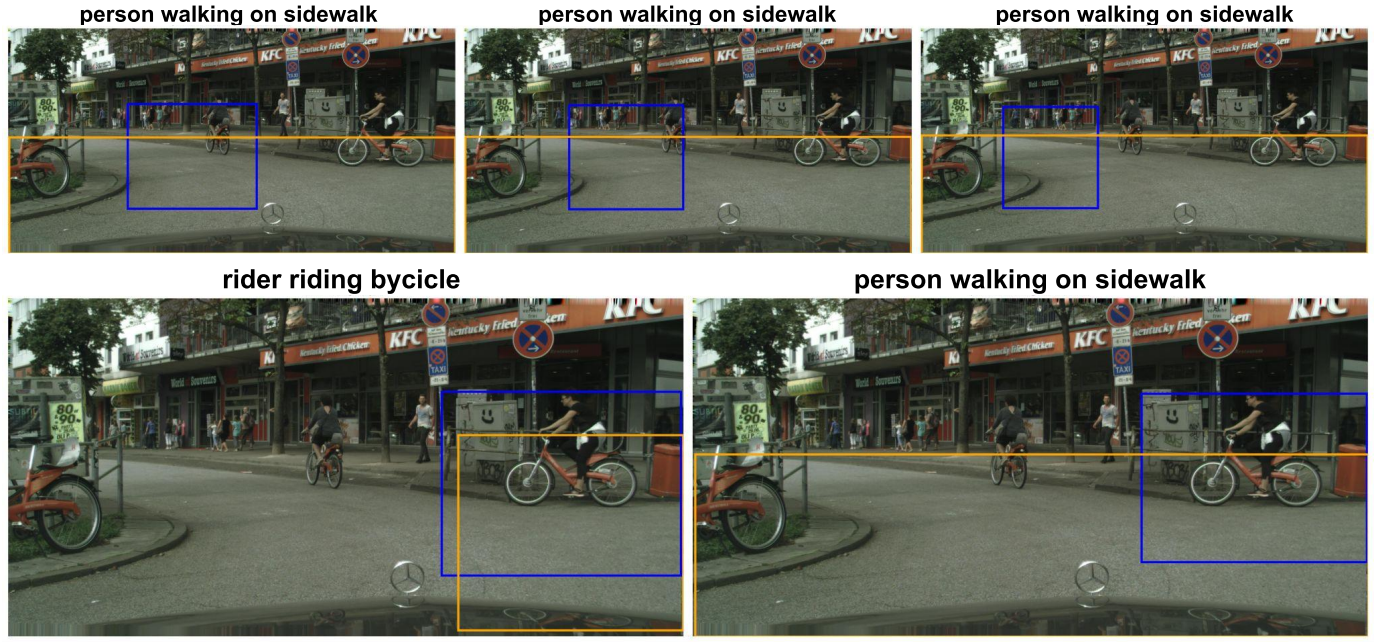
When classes such as *riding on* occur in busy scenes (Figure 4c), ESRA demonstrates a better ability to predict these classes. In addition, RelTR has difficulty properly separating objects in crowded environments, a challenge that ESRA does not have; on the other hand, ESRA tends to predict slightly larger bounding boxes.



(a). SGG of a sub-urban scene: (top), ESRA (bottom)



(b). SGG of a city scene: RelTR (top), ESRA (bottom)



(c). SGG of crowed traffic scene with tailed realtionsips: (top), ESRA (bottom)

Fig. 4: SGG outputs for both RelTR (top rows) and ESRA (bottom rows). Subject bounding boxes are represented in blue and object bounding boxes in yellow. Although both methods are capable of handling relationships that are not tailed (a, b), ESRA achieves better performances for tailed relationships (c).

VI. Conclusion

In this study, the feasibility of using LTNs in end-to-end transformers for SGG in the field of autonomous driving was investigated for the first time. Transformer models require large amounts of training data, but incrementing the size of the dataset on itself may not be sufficient to achieve good performance in real-world scenarios, as was shown. This emphasizes the essential role of prior knowledge, which has been shown to improve performance more prominently when providing information that is complementary to the original dataset annotations. Future work could test the development or conversion of external knowledge bases, such as open-source commonsense knowledge graphs, bypassing the need of manual annotations. Additionally, comparison or possible integration with other knowledge injection techniques could be explored, such as those based on knowledge graph embedding, and novel constraint formulations that could better integrate external information could be investigated.

Acknolewdgement

This study was carried out within the FAIR - Future Artificial Intelligence Research and received funding from the European Union Next-GenerationEU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) – MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.3 – D.D. 1555 11/10/2022, PE00000013). This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them. Computational resources were partially provided by HPC@POLITO, a project of Academic Computing within the Department of Control and Computer Engineering at Politecnico di Torino (http://www.hpc.polito.it).

References

[1] X. Ma, W. Ouyang, A. Simonelli, and E. Ricci, "3D object detection from images for autonomous driving: A survey," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 01, pp. 1–20, 2023.

[2] Y. Huang, J. Du, Z. Yang, Z. Zhou, L. Zhang, and H. Chen, "A survey on trajectory-prediction methods for autonomous driving," *IEEE Trans. on Intelligent Vehicles*, vol. 7, no. 3, pp. 652–674, 2022.

[3] K. Muhammad, T. Hussain, H. Ullah, J. Del Ser, M. Rezaei, N. Kumar, M. Hijji, P. Bellavista, and V. H. C. de Albuquerque, "Vision-based semantic segmentation in scene understanding for autonomous driving: Recent achievements, challenges, and outlooks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, 2022.

[4] X. Chang, P. Ren, P. Xu, Z. Li, X. Chen, and A. Hauptmann, "A comprehensive survey of scene graphs: Generation and application," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 1–26, 2021.

[5] Y. Cong, M. Y. Yang, and B. Rosenhahn, "RelTR: Relation transformer for scene graph generation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 9, pp. 11 169–11 183, 2023.

[6] H. Wang, T. Li, Y. Li, L. Chen, C. Sima, Z. Liu, B. Wang, P. Jia, Y. Wang, S. Jiang *et al.*, "OpenLane-v2: A topology reasoning benchmark for unified 3D HD mapping," vol. 36, 2024.

[7] T. Li, L. Chen, H. Wang, Y. Li, J. Yang, X. Geng, S. Jiang, Y. Wang, H. Xu, C. Xu, J. Yan, P. Luo, and H. Li, "Graph-based topology reasoning for driving scenes," *arXiv preprint arXiv:2304.05277*, 2023.

[8] A. V. Malawade, S.-Y. Yu, B. Hsu, H. Kaeley, A. Karra, and M. A. Al Faruque, "Roadscene2vec: A tool for extracting and embedding road scene-graphs," *Knowledge-Based Systems*, vol. 242, p. 108245, 2022.

[9] J. Wang, A. V. Malawade, J. Zhou, S.-Y. Yu, and M. A. Al Faruque, "RS2G: Data-driven scene-graph extraction and embedding for robust autonomous perception and scenario understanding," in *Proc. of the IEEE/CVF Winter Conf. on Applications of Computer Vision*, 2024, pp. 7493–7502.

[10] P. Kochakarn, D. De Martini, D. Omeiza, and L. Kunze, "Explainable action prediction through self-supervision on scene graphs," in *2023 IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2023, pp. 1479–1485.

[11] Z. Zhang, C. Zhang, Z. Niu, L. Wang, and Y. Liu, "GeneAnnotator: A semi-automatic annotation tool for visual scene graph," *arXiv preprint arXiv:2109.02226*, 2021.

[12] K. Tang, Y. Niu, J. Huang, J. Shi, and H. Zhang, "Unbiased scene graph generation from biased training," in *2020 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 3713–3722.

[13] I. Donadello and L. Serafini, "Compensating supervision incompleteness with prior knowledge in semantic image interpretation," in *2019 Int. Joint Conf. on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–8.

[14] S. Badreddine, A. d. Garcez, L. Serafini, and M. Spranger, "Logic tensor networks," *Artificial Intelligence*, vol. 303, p. 103649, 2022.

[15] F. Manigrasso, F. D. Miro, L. Morra, and F. Lamberti, "Faster-LTN: A neuro-symbolic, end-to-end object detection architecture," in *Artificial Neural Networks and Machine Learning – ICANN 2021*, I. Farkaš, P. Masulli, S. Otte, and S. Wermter, Eds. Cham: Springer Int. Publishing, 2021, pp. 40–52.

[16] L. Serafini and A. S. d'Avila Garcez, "Learning and reasoning with logic tensor networks," in *Conf. of the Italian Association for Artificial Intelligence*. Springer, 2016, pp. 334–348.

[17] S. Martone, F. Manigrasso, F. Lamberti, and L. Morra, "Prototypical logic tensor networks (PROTO-LTN) for zero shot learning," in *2022 26th Int. Conf. on Pattern Recognition (ICPR)*, 2022, pp. 4427–4433.

[18] A. Daniele and L. Serafini, "Knowledge enhanced neural networks for relational domains," in *AIxIA 2022 – Advances in Artificial Intelligence*, A. Dovier, A. Montanari, and A. Orlandini, Eds. Cham: Springer Int. Publishing, 2023, pp. 91–109.

[19] R. Krishna, Y. Zhu, O. Groth, and et al., "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *Int. Journal of Computer Vision*, vol. 123, pp. 32–73, 2017.

[20] J. D. Lafferty, A. McCallum, and F. C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. of the 18th Int. Conf. on Machine Learning*, 2001, pp. 282–289.

[21] D. Yu, B. Yang, Q. Wei, A. Li, and S. Pan, "A probabilistic graphical model based on neural-symbolic reasoning for visual relationship detection," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 10 609–10 618.

[22] T. Chen, W. Yu, R. Chen, and L. Lin, "Knowledge-embedded routing network for scene graph generation," in *Conf. on Computer Vision and Pattern Recognition*, 2019.

[23] Z. Chen, S. Rezayi, and S. Li, "More knowledge, less bias: Unbiasing scene graph generation with explicit ontological adjustment," in *2023 IEEE/CVF Winter Conf. on Applications of Computer Vision (WACV)*, 2023, pp. 4012–4021.

[24] L. Li, J. Xiao, H. Shi, W. Wang, J. Shao, A.-A. Liu, Y. Yang, and L. Chen, "Label semantic knowledge distillation for unbiased scene graph generation," *IEEE Trans. on Circuits and Systems for Video Technology*, 2023.

[25] Y. Zhang, Z. Liu, and S. Wang, "SrTR: Self-reasoning transformer with visual-linguistic knowledge for scene graph generation," *arXiv preprint arXiv:2212.09329*, 2022.

[26] X. Chang, T. Wang, S. Cai, and C. Sun, "Landmark: Language-guided representation enhancement framework for scene graph generation," *Applied Intelligence*, vol. 53, no. 21, pp. 26 126–26 138, 2023.

[27] J. Zhang, K. J. Shih, A. Elgammal, A. Tao, and B. Catanzaro, "Graphical contrastive losses for scene graph parsing," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, 2019, pp. 11 535–11 543.

[28] K. Yoon, K. Kim, J. Moon, and C. Park, "Unbiased heterogeneous scene graph generation with relation-aware message passing neural network," in *Proc. of the AAAI Conf. on Artificial Intelligence*, vol. 37, no. 3, 2023, pp. 3285–3294.

[29] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European Conf. on computer vision*. Springer, 2020, pp. 213–229.

[30] H. Li, G. Zhu, L. Zhang, Y. Jiang, Y. Dang, H. Hou, P. Shen, X. Zhao, S. A. A. Shah, and M. Bennamoun, "Scene graph generation: A comprehensive survey," *Neurocomputing*, vol. 566, p. 127052, 2024.