Probing Spatial Clues: Canonical Spatial Templates for Object Relationship Understanding

GUILLEM COLLELL, THIERRY DERUYTTERE, AND MARIE-FRANCINE MOENS
Computer Science Department, KU Leuven, 3001 Leuven, Heverlee, Belgium
Corresponding author: Guillem Collell (collellguillem@gmail.com)

This work was supported in part by the European Research Council (ERC) Advanced Grant Commonsense and Anticipation enriched Learning of Contain/ious representations supporting Language UnderStanding (CALCULUS) under Grant H2020-ERC-2017-ADG 788506, and in part by the Fonds Wetenschappelijk Onderzoek (FWO)-Swiss National Science Foundation (SNSF) Grant DEEP LEARNING FOR GENERATING TEMPLATE PICTORIAL AND TEXTUAL REPRESENTATIONS (DEEPTEMPL) under Grant FWO-SNSF G078618N. The work of Thierry Deruyttere was supported by the Artificial Intelligence (AI) Flanders Program.

ABSTRACT
Humans often leverage spatial clues to categorize scenes in a fraction of a second. This form of intelligence is very relevant in time-critical situations (e.g., when driving a car) and valuable to transfer to automated systems. This work investigates the predictive power of solely processing spatial clues for scene understanding in 2D images and compares such an approach with the predictive power of visual appearance. To this end, we design the laboratory task of predicting the identity of two objects (e.g., “man” and “horse”) and their relationship or predicate (e.g., “riding”) given exclusively the ground truth bounding box coordinates of both objects. We also measure the performance attainable in Human Object Interaction (HOI) detection, a real-world spatial task, which includes a setting where ground truth boxes are not available at test time. An additional goal is to identify the principles necessary to effectively represent a spatial template, that is, the visual region in which two objects involved in a relationship expressed by a predicate occur. We propose a scale-, mirror-, and translation-invariant representation that captures the spatial essence of the relationship, i.e., a canonical spatial representation. Tests in two benchmarks reveal: (1) High performance is attainable by using exclusively spatial information in all tasks. (2) In HOI detection, the canonical template outperforms the rest of spatial, visual, and several state-of-the-art baselines. (3) Simple fusion of visual and spatial features substantially improves performance. (4) Our methods fare remarkably well with a small amount of data and rare categories. Our results obtained on the Visual Genome (VG) and the Humans Interacting with Common Objects - Detection (HICO-DET) datasets indicate that great predictive power can be obtained from spatial clues alone, opening up possibilities for performing fast scene understanding at a glance.

INDEX TERMS
Spatial understanding, spatial layout, computer vision, vision and scene understanding.

I. INTRODUCTION
A well-researched concept in cognitive science is the gist, or the initial representation of a scene obtained in a brief glance. The gist may include semantic content (e.g., “is a classroom”), the identity of a few objects (e.g., “there are books”), and the spatial layout [1]. Humans can categorize scenes in a fraction of a second (∼13-250 ms) [1], [2]. Generally, more detailed scenes and finer-grained judgments require examining the scene for a longer time [3], [4]. To perform such fast scene categorization, humans leverage a small subset of scene descriptors, including spatial clues, context, and semantic properties of objects [1], [5]. Additionally, depth can also be leveraged for certain relations such as “before” or “behind”. A large body of research shows that spatial information is a strong clue for fast scene categorization, including the spatial dependency between objects [1], [6], the objects’ relation to the scene layout [7] and global spatial properties such as recognizing a “street” because it is an open space flanked with tall vertical surfaces (i.e., buildings) [2].

In addition, vast evidence from Artificial Intelligence (AI) shows that explicitly modeling and integrating spatial information into a system can provide a performance boost over
its non-spatial ablated counterparts in several tasks, including Human Object Interaction (HOI) detection [8], Concept Similarity [9], and Object Recognition [10]. It is also known that neural transformer architectures that integrate multi-head attention mechanisms implicitly can learn spatial relations among a multitude of other relationships (e.g., [11]).

However, the predictive power of using exclusively spatial information in scene understanding tasks with 2D images has not been systematically studied yet. In time-sensitive settings (e.g., scene understanding by a self-driving car), spatial cues might provide a reliable first filtering of the many possible objects and their actions that the car could encounter in the real world and speed up the detection process. Our first research question (RQ 1) is: what portion of scene understanding can be credited to the spatial component alone? To this end, we design a minimalist laboratory task (triplet categorization) consisting in predicting the identity of two objects (e.g., "man" and "dog") and their relationship (e.g., the predicate "walking") given exclusively the spatial layout of the objects (bounding box coordinates), and discarding pixel values. We also determine the performance attainable by using spatial models in the real-world task of HOI detection, a variant of object detection where both human and object boxes need to be correctly located in human-relation-object instances. Furthermore, in all tasks we systematically compare the predictive power of spatial clues versus visual appearance, i.e., state-of-the-art (SOTA) visual convolutional neural network (CNN) features.

Because spatial layouts are encoded with only a few real-values (coordinates), doing computations with them is remarkably fast (e.g., element-wise vector comparisons). Hence, if spatial clues alone prove a sufficiently strong predictor, promising possibilities would open up for their use as a fast diagnostic tool in real-time scene understanding applications [12], analogous to humans’ scene ‘gist’. Intuitively, e.g., given two equally-sized bounding boxes next to each other (horizontally), one can quickly rule out “man riding horse” as a plausible candidate category.

To make spatial knowledge actionable in real-world tasks we leverage spatial templates, a well-established framework in both the AI [13]–[15] and cognitive science [16]–[19] communities. Given two objects under a relationship (e.g., “on” or “riding”), a spatial template computes both objects’ expected regions of acceptability. For instance, given (man, riding, horse), one would expect the man on the horse and not next to it.

Our second research question (RQ 2) is to disentangle the principles necessary for effectively representing spatial templates.

To this end, we systematically evaluate the effect of applying principled geometric transformations (e.g., centering) with different capabilities (translation-, scale-invariance, etc.) to the original spatial layouts or bounding boxes. Although the role of coordinate systems on human spatial template acquisition and mental representation has been thoroughly studied in psychology [16]–[18], similarly exhaustive studies are lacking in scene understanding.

In particular, we propose a spatial representation that is scale-, mirroring- and translation-invariant. We name this representation as a canonical template in virtue of capturing only the spatial essence of an object-relation-object triplet, while discarding all the information that is arbitrary and irrelevant towards understanding the scene, such as re-scalings, translations or image horizontal-flips (mirroring). In contrast with, for instance, latent features, the canonical template is interpretable and can be visualized in a 2D space (e.g., Fig. 4).

We provide methods to make spatial templates actionable in real-world tasks. In addition, we compare a variety of learning methods to learn spatial templates from the transformed layouts, including neural networks, statistical and decision tree classifiers.

We conduct extensive tests in two benchmark datasets. Our results show that: (1) RQ 1: High performance is
attainable by using exclusively spatial information in both, the purely semantic task (triplet categorization) and the task with spatial emphasis (HOI detection). All spatial representations considered provide good predictive power, well-over chance level. (2) **RQ 2**: In the spatially focused task, HOI detection, the spatial canonical representation outperforms the rest of spatial methods, some SOTA systems that use pixel values, and visual appearance (i.e., CNN features) while being several orders of magnitude faster than CNN. (3) The simple early and late fusion of our spatial methods and CNN features provides a performance boost w.r.t. each method alone in all tasks. This suggests that spatial clues and visual appearance are not redundant in images. Therefore substantial improvements in both semantically- and spatially-focused scene understanding tasks can be seized by explicitly modelling both sources of information in a system. (4) A simple model using category prototypes outperforms most state-of-the-art HOI detection methods in rare categories. This simple model represents each category (prototype) as the average of the feature vectors of all training samples for that category. An extra experiment where we heavily undersample the training set further supports this point and reveals that ten samples per category suffice to attain performances close to those using the full data set. Overall, the low computational cost and high predictive power of principled canonical representations support their potential use as a compact scene descriptor for applications that require fast real-time scene understanding, such as realized by self-driving cars [12].

The rest of the paper is organized as follows. In Sect. II we introduce core concepts. In Sect. III we discuss related research. In Sect. IV we define spatial layout transformations (Sect. IV-A1) and the learning methods (Sect. IV-B1). In Sect. V we describe our two tasks. In Sect. VI we describe the experimental setup and datasets employed. In Sect. VII we present and discuss the empirical results. In Sect. VIII we conclude and suggest future work.

II. PRELIMINARIES

A. SPATIAL LAYOUTS AND TEMPLATES (DEFINITIONS)

Let us consider a vocabulary of $m$ objects $O = \{O_1, \ldots, O_m\}$ (e.g., “cat”, “horse”, etc.) and a vocabulary of $k$ Relations $\mathcal{R} = \{R_1, \ldots, R_k\}$. The Relationships can be either spatial prepositions (“on”, “below”, etc.) or actions (“riding”, “throwing”, etc.).

The basic working unit in this article is relationship-object *triplets*\(^1\)* of the form $t = (S, R, O) \in T \subseteq O \times \mathcal{R} \times O$, which stands for (Subject, Relationship, Object), and $T$ denotes our set of triplets or triplet vocabulary. Here, the Subject is not necessarily a syntactic subject but simply the referent object, which also happens to be the syntactic subject when the Relation is an action. We henceforth capitalize the Object when it refers to the Object, $O$, of a triplet $t = (S, R, O)$, and we do not capitalize it when we refer to objects in the traditional sense of the word. Similarly for Relations. Notice that $(S, R, O)$ triplets are general enough to encode any spatial interaction. This includes the case where two objects simply co-exist in a scene without exhibiting any interaction. In that case one encodes: (Subject, no interaction, Object).

Given a particular instance of a triplet class $t = (S, R, O) \in T$, we define its spatial layout $S_t$ as the pair

\[ s_t = [b^S, b^O] \quad (1) \]

consisting of the Subject’s bounding box $b^S$ and the Object’s box $b^O$. Boxes are parametrized by

\[ b^O = [x^O_1, y^O_1, h^O, w^O]; \quad b^S = [x^S_1, y^S_1, h^S, w^S] \quad (2) \]

where $x^O_1, y^O_1$ stand for the $x$ and $y$ coordinates of the Object center respectively, and $h^O, w^O$ are its height and width respectively. Similarly for the Subject $b^S$. See rightmost sketch of Fig. 3 for notation. For simplicity, we use the term spatial layout to refer to any way of encoding or parametrizing such boxes, including any transformation $\phi$ applied to them: $\hat{s}_t = \phi(s_t)$ (Sect. IV-A).

Intuitively, the **spatial template** $\hat{s}_t$ of a triplet class $t = (S, R, O)$ is its expected (set of) spatial layout(s), i.e., the plausible spatial arrangements of $S$ and $O$. This corresponds to the regions of acceptability for both Object $O$ and Subject $S$ (referent), given their relationship $R$. Formally, the spatial template of $t = (S, R, O)$ is a specification of the likelihood that the points $p_S, p_O$ are a part of the $S$’s and $O$’s location, respectively, in a 2D space. E.g., a simple and deterministic template for $t = (S, R, O)$ may be two bounding-box-like distributions $(b^S, b^O)$ with probability 1 of localizing $S$ and $O$ inside each respective box and probability 0 outside [13]. A more refined template may be the pixel-wise averaged box across all available instances (layouts) of a triplet class $t = (S, R, O)$ (as in Fig. 4). Spatial templates will be learned from the different transformed spatial layouts (Sect. IV-A1).

Notice the distinction we make between the term **template** which refers to an expected, learned spatial arrangement (e.g., a mental model) for a given triplet class $t = (S, R, O)$, and the term **layout** which refers to the concrete, observed spatial arrangement in an individual instance $t_i$ of triplet class $t$ (e.g., in a particular image). Typically, the purpose of learning templates is to enable computing the match between a particular observed spatial layout and the expected spatial template of some candidate triplet class $t$.

B. TYPES OF SPATIAL LAYOUT VARIABILITY

For an instance of a triplet class $t = (S, R, O)$, there are often multiple plausible spatial layouts that the objects may exhibit

\(^1\) Notice that $(S, R, O)$ triplets can be automatically extracted from natural language with a parser [20], currently with a Labeled attachment scores (LAS) of over 88%.

\(^2\) Other box parametrizations are possible, such as $b^O = [x^O_1, y^O_1, x^O_2, y^O_2]$ with $(x^O_1, y^O_1)$ the top left corner and $(x^O_2, y^O_2)$ the lower right corner of $O$’s box. Similarly for $b^S$. The (rather minor) effect of re-parametrizations on results is discussed in the Supplement.
in 2D natural images. We distinguish between two sources of variability: **extrinsic** and **intrinsic**, which respectively does or does not depend on camera location.

**Extrinsic:** (i) **Image crops and translations:** Jointly translating a pair of objects or cropping part of an image does not modify the scene’s meaning. (ii) **Re-scaling:** Re-scaling images or object pairs do not alter meaning. (iii) **Horizontal flips:** A mirrored image preserves its meaning [13], yet exhibits a different layout. (iv) **Camera viewpoint:** That is, the variability in 2D spatial layouts that stems from placing the camera at a different location in a 3D world.

**Intrinsic:** (i) **Location flexibility:** Some scenes intrinsically admit more flexibility than others due to the very nature of the action and objects involved. E.g., given (man, walking, horse) the “man” can plausibly be either at the front, the back or the lateral of the “horse”. However, given (man, riding, horse), the man can only be on top of the horse. Hence, the spatial layout of (man, walking, horse) is inherently more flexible than that of (man, riding, horse). (ii) **Pose:** E.g., a “man cutting apple” may be either standing or sitting, which will affect the spatial layout configuration.

Ideally, the learned templates would discount / ignore extrinsic variability to the extent possible.

### III. RELATED WORK

#### A. RELATED TASKS

1) **OBJECT RECOGNITION**

The task of object recognition [21] consists in labeling the objects present in images. Shiang et al. [10] show that performance in object recognition can be boosted by incorporating spatial contextual knowledge of objects. The authors learn object-object relative locations and their co-occurrences from external databases and leverage this knowledge to rank candidate object labelings.

2) **ACTION RECOGNITION**

The goal of the action recognition task is to recognize (classify) which action a human (Subject) is performing with an Object, given an image as input [22]–[24]. A common way [23], [25], [26] to handle this task is by estimating the body pose of the person performing an action. For this Maji et al. [23] use body part detectors, or poselets [27]. A poselet activation vector is then used to classify the action. Instead of predicting the pose directly, Hu et al. [25] use HOI descriptors indicating the spatial relation between Subject and Object, as well as human pose. The action is then classified based on this descriptor. Note that this task is different from HOI detection [8] as action recognition does not evaluate how well the objects’ locations are predicted.

3) **OBJECT DETECTION**

The object detection task consists in labeling and localizing objects in an image, and has been extensively researched [28]–[31]. In this work, we use one of the most well-known object detection networks, Faster-RCNN [28]. Reference [28] improves [29] by using a Region Proposal Network (RPN), which proposes the regions based on the extracted feature map by a CNN for a given image.

4) **VISUAL RELATIONSHIP (VR) DETECTION**

A visual relationship is composed of a predicate and its context, that is Subject and Object. Although VR detection is a well-studied problem (e.g., Lu et al. [32], Liang et al. [33]) and many works exploit spatial features to accomplish this task (e.g., [34]–[37]), few works (e.g., [38]) predict visual relations and localize them in the visual scene.

In this paper, we limit VR detection to HOI detection, which requires the correctness of the triplet class as well as the Subject and Object box locations. Hence, HOI detection [8] is an explicitly spatial task.

5) **HUMAN OBJECT INTERACTION (HOI) DETECTION**

Chao et al. [8] introduce the detection version of the HOI recognition task [39], [40]. While in HOI recognition, the goal is to predict correctly the triplet classes (S, R, O) present in an image. In HOI detection, one must additionally predict the location of both correctly, Subject and Object boxes, i.e., each box must have an overlap larger than 50% with its corresponding ground-truth box. To date, a body of work have approached the HOI detection problem [41]–[55]. Several of these works do not explicitly integrate spatial information with regard to position, size, or layout of the involved human and objects (e.g., [53]), or integrate this information or part of it in a non-transparent way in the neural network (e.g., [46], [49], [50]). Among the latter are also approaches that model spatial layouts by means of interaction patterns that characterize the relative location of two bounding boxes (e.g., [8], [41], [44], [47], [51]). Given a pair of bounding boxes, its interaction pattern is a binary image with two channels: The first channel has value 1 at pixels enclosed by the first bounding box, and value 0 elsewhere; the second channel has value 1 at pixels enclosed by the second bounding box, and value 0 elsewhere. Reference [43] propose a neural network that integrates three factors: visual appearance, human pose, and layout of object boxes, and the ablation study showed that all three properties contributed to solving the HOI detection task. [43], [44], [51] and [46] show that pose information forms an additional cue. Gkioxari et al. [52] propose a deep end-to-end architecture that integrates three branches: a region proposal network (identical to Faster-R-CNN), a human-centric branch, and a human-object interaction branch. Their human-centric model assumes that the person’s appearance (clothing, pose, etc.) is a strong clue for localizing the object. Hence they predict a spatial distribution for the target object based on the detected person’s appearance. Kato et al. [56] leverage external resources such as WordNet in HOI detection. Peyre et al. [45] focus on learning object and predicate embeddings separately as well as a joint embedding for the combined triplet. They also learn how to transfer the joint embedding from seen triplets to unseen ones, enabling generalization to rare triplets. Bansal et al. [57]
leverage the common-sense knowledge that humans have similar interactions with objects that are functionally similar. They automatically discover functionally similar objects by relying on word embeddings, appearance features, and spatial information of the bounding boxes of the object involved. However, the goal of their work is different from ours.

In contrast to all the above works, the main focus of this paper explicitly measures the contribution of spatial clues when solving the HOI detection task. Hence, we simplify our experimental design to the possible extent and build our models with the question “what is the simplest approach?” in mind, rather than “how can we improve performance?”. Nevertheless, our simple models attain performances competitive with some deep network approaches while being fully interpretable and computationally cheap.

6) IMAGE GENERATION
Johnson et al. [58] do text-to-image generation with an adversarial network approach. They take a multi-object scene graph as input (i.e., a collection of \((S, R, O)\) triplets) and predict the 2D pixel canvas of an image. As an intermediate step between input and output, they predict the bounding boxes of all objects as a regression task to place the objects in the 2D space. Zitnick et al. [59] model the generation of images (and text-based image retrieval) using a Conditional Random Field (CRF) using the objects and their relations (including spatial locations) as nodes and edges, respectively. In contrast with ours, the focus of these works is not on spatial prediction, and they employ absolute coordinates for objects rather than relative. We compare with absolute coordinates akin to theirs in our experiment (Sect. IV-C).

In the context of scene graph generation [60] derive counterfactual causality from the trained graph to infer the effect of the bad bias and then remove it. Their evaluation condition SGCIs is similar to our triplet categorization task as they also use ground truth boxes. However, they use visual information (while we do not) and they employ a retrieval setting – hence they use mean Recall at K – ours is a classification setting.

7) SPATIAL-BASED IMAGE CAPTIONING
Yin and Ordonez [61] generate captions for images using a set of objects and their spatial locations exclusively. Their sequence-to-sequence model (OBJ2TEXT) encodes the objects and locations using an LSTM and outputs a text sequence using an LSTM decoder. In contrast, we use images as input and decode the object-relation-object categories, which can be seen as structured language.

8) IMAGE RETRIEVAL (SPACE-FOCUS)
Mai et al. [62] propose a CNN model to perform image search from text queries with spatial constraints specified by the user. The user first places multiple text-boxes associated with objects on a 2D query canvas, and then the system retrieves images with a similar spatial layout. The purpose of our work is rather the contrary, i.e., instead of having to manually specify the layout for a given text query (e.g., a \((S, R, O)\) triplet), we aim to learn its expected layout (e.g., a template). Malinowski and Fritz [15] use spatial templates to retrieve images given \((S, R, O)\) textual queries, yet restricting the templates to explicit spatial prepositions (e.g., “above”, etc.).

B. RESEARCH ON SPATIAL TEMPLATES
The term spatial template was introduced by Logan and Sadler [19] to describe the regions of acceptability associated with a spatial preposition (e.g., “on” or “below”). Since then, spatial templates have been extensively studied in the cognitive science [16]–[18] as well as in the machine learning literature [13], [15], [63], [64].

Early machine learning approaches to spatial templates were rule-based [64], i.e., the regions of acceptability were hand-coded (e.g., the template for “left” comprehends angles between 30° and 160°, etc.). More recent work considers learning approaches. Malinowski and Fritz [15] learn the parameters of spatial templates with a pooling operation and leverage them to retrieve images given text queries, which as mentioned explicitly express a spatial relationship signaled by a spatial preposition (e.g., “on”). The authors compute the soft spatial fit between a template and the observed object boxes. For instance, a box to the right-hand side of the referent object gets a low score for the template “left of”, but a high score for “right of.”

Collell et al. [13] extended the concept of spatial templates from explicit spatial prepositions (e.g., “on” or “below”) to implicit spatial Relations (i.e., actions such as “riding” or “pulling”). Interestingly, [13] find that spatial templates for implicit Relations can be as accurately predicted as for explicit Relations. In contrast with [13] whose goal is the prediction and evaluation of spatial templates per se, our goal is to make spatial templates actionable in real-world tasks. Additionally, the authors of [13] do not focus on discussing layout transformations.

In contrast with the prior work that restricts to explicit spatial prepositions [15]–[19], we learn a template for each triplet \(t = (S, R, O)\) as in [13], instead of a template for each Relation \(R\). Thus, we allow higher flexibility, at the expense of incurring higher sparsity. However, as noted by [13], templates for implicit spatial Relations (actions) are generally not as “rigid” as explicit spatial prepositions (e.g., “on”, “above”) and hence cannot be treated equally. An explicit spatial Relation (e.g., “on”) generally always suffices to determine the objects’ location. That is, (Subject, on, Object) always implies that the Subject is on top of the Object. However, many implicit Relations (e.g., “wearing”) do not tell much about the objects’ locations (unless we know the objects involved). That is, while in (man, wearing, shoes) the “shoes” (O) are at the bottom of the “man” (S), in (man, wearing, hat) the “hat” (O) is at the “man”’s top (S), and nevertheless the Relation is the same (“wearing”). Hence the need to consider the triplet \((S, R, O)\) as an atomic template unit, rather than the Relation \(R\) alone.
Moreover, triplets \((S, R, O)\) provide the proper level of granularity to enable: (1) easily making templates actionable in tasks; (2) easily visualizing the effect of different representational choices (e.g., scale invariance); and (3) compatibility with a myriad of tasks that use \((S, R, O)\) triplets at their core, such as: “spatial question answering” [13], HOI recognition [25], [40], [65], HOI detection [8], action recognition [23], visual relationship detection [32], image generation (text-to-image synthesis) [58], 3D text-to-scene generation [14], extraction of common sense facts [66], image retrieval [15], [32], image annotation [15], or image description [67].

C. COORDINATE SYSTEMS IN SPATIAL TEMPLATES

A large body of research in the psychological literature has exhaustively compared the role of different coordinate systems on human mental representation and comprehension of spatial templates [16]–[19]. For instance, the Bounding Box model employs Cartesian coordinates, whereas the Proximal and Center of Mass model assumes polar coordinates, namely the angle between the trajector (equivalent to our Subject) and the landmark (Object) [17]. In fact, the term canonical has been previously employed in the psychological literature to denote the spatial templates exhibiting a more “natural” appearance (e.g., non-rotated) [18].

However, a similarly exhaustive analysis of coordinate systems to represent spatial templates lacks automatic scene understanding. Zhang et al. [38] propose a method for scale-invariant translation of spatial features used in visual relationship extraction. A common trend is to use off-the-shelf black-box models to automatically learn (a sequence of) appropriate transformations to an input pair of object boxes that best serves in a task. However, their combination with principled methods presents some major advantages. In particular, principled geometric representations: (1) do not need training data; (2) are interpretable (in contrast with, e.g., latent features); (3) are hyperparameter-free (hence no extra data for tuning); and (4) are cheap to compute. Such advantages make them suitable in scenarios where either interpretability or speed are key, or training data are unavailable or scarce. Canonical templates do not exclude the use of deep neural networks to learn spatial knowledge but rather strengthens and complements them by gaining insight on the principles needed for representing spatial templates effectively. Ultimately, principled inductive biases may relieve a model from implementing a large number of parameters devoted to learning intuitive transformations that are naturally needed for a task (e.g., scale invariance).

IV. PROPOSED METHODS

In this section, we give a general overview of our proposed spatial layout representations as well as the used methods. In the experiment section, we combine some of these methods.

A. SPATIAL LAYOUT REPRESENTATIONS

To address our second research question (RQ 2), we consider different layout transformations \(\phi(s_i)\), which yield actionable representations for our tasks. The goal is to find a canonical template that captures the spatial essence of a given triplet class \(t = (S, R, O)\).

1) SPATIAL LAYOUT TRANSFORMATIONS

a: IMAGE PRE-PROCESSING

To preserve the aspect ratios of the objects when using normalized coordinates, we first “pad” the shortest side (either height or width) of the spatial image (i.e., only coordinates, without pixel values) with empty space and subsequently normalize (Fig. 2). That is, \(b^S = \frac{1}{\alpha} b^S_{\text{pix}}\) where \(b^S_{\text{pix}}\) is the Subject’s \((S)\) bounding box in pixel integer coordinates and \(\alpha = \max(h^\text{img}, w^\text{img})\) with \(h^\text{img}\) and \(w^\text{img}\) the height and width of the image, respectively. Similarly for \(b^O = \frac{1}{\alpha} b^O_{\text{pix}}\).

![FIGURE 2. Spatial image pre-processing. Red bounding box indicates the object O while the blue bounding box indicates the subject S.](image)

The transformations \(\phi(s_i)\) applied onto the above normalized layout \(s_i = \{b^S; b^O\} \in [0, 1]^8\) are detailed below.

b: MIRROiNG

The mirroring or horizontal flip operation \(M()\) (last step in Fig. 3) is defined as:

\[
M(c^S_x, c^O_x) = \begin{cases} 
(1 - c^S_x, 1 - c^O_x) & \text{if } c^O_x < c^S_x \\
(c^S_x, c^O_x) & \text{Otherwise}
\end{cases}
\]

Intuitively, \(M()\) mirrors the image only when the Object is located at the left of the Subject, leaving as a result the Object always to the right-hand-side of the Subject, avoiding thus the left/right arbitrariness in images [13]. Please note that this operation is not appropriate in relations such as “left of” or “right of”.

c: INTERSECTION OVER UNION (IoU)

We consider adding a scalar value representing the overlap between the Subj and Obj bounding boxes as an extra spatial feature. This overlap will be referred to as Intersection over Union (IoU), and it will be concatenated to the given layout: \(s_i = \{b^S; b^O; \text{IoU}\}\), as in [43], for instance. The IoU equals to \(|b^S \cap b^O|/|b^S \cup b^O|\), with \(|b^S \cap b^O|\) the area of the intersection of Subj and Obj boxes, and \(|b^S \cup b^O|\) their union box’s area.

d: CENTERING

Centering the coordinates of \(O\) w.r.t. the \(S\) amounts to making the coordinates relative. The goal of including centered
layouts is to determine whether relativity alone (i.e., translation invariance) is sufficient or rather additional capabilities are needed (e.g., scale invariance). The centering (ctr) operation $C()$ is:

\[
\begin{align*}
    c_x^O & := 0.5 + x^O - c_x^S \\
    c_y^O & := 0.5 + y^O - c_y^S \\
    c_x^S & := 0.5 \\
    c_y^S & := 0.5
\end{align*}
\]

Then, if the translated Object box $b^O$ falls outside the original $[0, 1] \times [0, 1]$ box, we pad the largest excess, say $p$ units (from whichever dimension has the largest excess, i.e., $p = \max(\text{horizontal excess}, \text{vertical excess})$), symmetrically to each of the four sides of the original square $[0, 1] \times [0, 1]$ in order to get a square canvas again, now of size: $[−p/2, 1 + p/2] \times [−p/2, 1 + p/2]$. We re-normalize all coordinates by the new width and height produced by this padding, to re-gain $[0, 1] \times [0, 1]$ coordinates.

**e: CROP-PAD-NORMALIZE (CPN)**

We propose the Crop-Pad-Normalize (CPN) layout (Fig. 3), which consists in:

1) **Crop** the tightest box (“TB”) that contains both $b^S$ and $b^O$, delimited by:

\[
\begin{align*}
    y_{\text{TB}}^\text{low} & = \max(c_x^S + h^S/2, c_x^O + h^O/2) \\
    y_{\text{TB}}^\text{top} & = \min(c_x^S - h^S/2, c_x^O - h^O/2) \\
    x_{\text{left}} & = \min(c_x^S - w^S/2, c_x^O - w^O/2) \\
    x_{\text{right}} & = \max(c_x^S + w^S/2, c_x^O + w^O/2)
\end{align*}
\]

Thus, its height is $h_{\text{TB}} = y_{\text{TB}}^\text{top} - y_{\text{TB}}^\text{low}$ and its width $w_{\text{TB}} = x_{\text{right}} - x_{\text{left}}$.

2) **Pad** symmetrically the smallest side (either height $h_{\text{TB}}$ or width $w_{\text{TB}}$) of the tightest box to get a square box of height and width: $D = h_{\text{TB}} = w_{\text{TB}} = \max(h_{\text{TB}}, w_{\text{TB}})$ (“P” ~ Padded). Hence the (0, 0) point or top left corner of the padded box is:

\[
(x_{\text{left}}^P, y_{\text{top}}^P) = \left(\frac{x_{\text{left}} + x_{\text{right}}}{2} - \frac{D}{2}, \frac{y_{\text{top}} + y_{\text{bottom}}}{2} - \frac{D}{2}\right)
\]

Padding preserves the objects’ aspect ratio when normalizing by the new width and height below.

3) **Normalize** coordinates by $D$. Normalizing by $D$ (only after cropping) provides scale invariance, i.e., ensures that HOIs far away from the camera have the same scale as HOIs that are close to the camera. Cropping ensures that Subject and Object are always “enclosed” identically inside the tightest box, and normalization (dividing by $D$) “resets” the scale of the axes.

The complete $s_{\text{CPN}}$ layout transformation when the input is $(c_x^S, c_y^S, h^S, w^S, c_x^O, c_y^O, h^O, w^O)$, is equal to:

\[
s_{\text{CPN}} = \left(\frac{c_x^S - x_{\text{left}}}{D}, \frac{c_y^S - y_{\text{top}}}{D}, h^S, w^S, \frac{c_x^O - x_{\text{left}}}{D}, \frac{c_y^O - y_{\text{top}}}{D}, h^O, w^O\right)
\]

Our CPN representation differs from that of [8] in two aspects. First, we use continuous coordinates instead of a 2D Boolean mask that indicates the object’s location. Second and most importantly, after cropping and padding, we re-normalize the coordinates according to the padded tightest box. This makes our representation scale-invariant – in contrast to that of [8].

Our proposed CPN representation is: Scale-invariant, translation-invariant, a relative coordinate system, preserves aspect ratios, interpretable and computationally cheap. Thus, the composition of CPN and mirror (CPN + mirror) further provides mirror-invariance. Among the spatial layouts considered, CPN + mirror is the closest representation to a canonical layout. Effectively, it maps layouts that are visually distinct (re-scaled, translated, mirrored, etc.) but are semantically similar (i.e., same $(S, R, O)$ class) to the same canonical layout. The only sources of within-triplet variability that CPN + mirror does not account for are the intrinsic ones (e.g., pose) and camera viewpoint variations (Sect. II-B).3

2) **QUALITATIVE INTUITION**

Here, we qualitatively inspect the effect of the layout transformations above (Sect. IV-A1). To this end, it is instructive to visualize the transformed spatial layouts averaged across all available samples (instantiations) of a given triplet class $t = (S, R, O)$. A successful transformation $\phi(t)$ should

3Note that being mirroring-invariant is appropriate for interaction-type predicates, but not for some spatial predicates (e.g., “left of”, “right of”), which might be important in spatial navigation.
map different instantiations of the same triplet class $t = (S, R, O)$ – which may exhibit a wide variety of spatial layouts – into a similar canonical representation (template) that groups them together. Thus, the averaged layouts should ideally show a clean and defined visual pattern rather than a diffuse or scattered one. Intuitively, this representation should be more effective when used as a spatial descriptor in real-world tasks.

Fig. 4 illustrates that mirroring alone generally provides a cleaner, de-blurred pattern w.r.t. the Original (absolute) spatial layout. By further centering the Object w.r.t. the Subject (Ctr + mirror), a cleaner spatial pattern can be discerned. However, centering alone does not provide scale invariance. In general, CPN + mirror tends to provide the neatest and most clearly defined spatial patterns (or templates).

B. METHODS

In HOI detection (Sect. VII-B), any system must necessarily possess some internal representation of how each triplet class $t = (S, R, O)$ looks like. This can be achieved either in a discriminative fashion by learning to classify spatial layouts into triplet classes (layout-classification methods, Sect. IV-B2); or by explicitly learning an expected spatial layout or spatial template for each triplet class $t = (S, R, O)$ (Fig. 5). Since their outputs differ, the mechanism used to make them actionable in our tasks will also differ (Sect. IV-B1 and IV-B2). In both cases, the match score $m(t, s)$ can be interpreted as a measure of visual similarity between the observed layout $s$ and the average spatial layout (i.e., template) of a triplet class $t = (S, R, O)$. A similar interpretation holds when templates consist of visual CNN-extracted features.

We consider a wide spectrum of learning methods encompassing: a simple baseline (layout averaging); statistical methods (LDA, QDA); neural network methods (feed-forward network); classical tree ensembles (random forest).

1) TEMPLATE-BASED METHODS

Given a triplet class $t = (S, R, O)$ as input, the model $f$ outputs its spatial template $\hat{s}_t = f(t)$, i.e., the expected layout

![Figure 4](image_url)

**FIGURE 4.** Averaged layouts across all available samples for each triplet class $(S, R, O)$ in VG dataset [68], with four layout transformations (columns). The Subject is shown in blue and the Object in red (best seen in color). Original denotes the initial bounding boxes without applying any transformation (Abs in Sect. IV-C); Mirror applies horizontal-flip (when needed) (Sect. IV-A1.b); Ctr + Mirr centers the Subj and translates the Obj accordingly (Sect. IV-A1.d); CPN + Mirr is the canonical layout (Sect. IV-A1.e). The number of samples of the averaged layouts above range between 27 and 1665.

![Figure 5](image_url)

**FIGURE 5.** Template-based (top) and Layout-classification (bottom) methods to compute the Layout-class match $m(t, s)$. Red bounding box indicates the object $O$ while the blue bounding box indicates the subject $S$. 
of triplet $t$. The shape of $\hat{s}_t$ is the same as the individual layouts $s_t$, to enable easily computing a layout-template element-wise vector match.

The layout-class match $m(t, s)$ between the observed layout $s$ and a given (candidate) triplet class $t$ is computed via cosine similarity\,\footnote{Other similarity measures such as the Euclidean similarity and mean square error were considered. Cosine fared slightly better.} between $s$ and the (learned) spatial template $\hat{s}_t$ for triplet class $t$ (Fig. 5, top):

$$m(t, s) = \cos(\hat{s}_t, s) = \frac{\hat{s}_t s}{\|\hat{s}_t\| \|s\|} \quad (10)$$

We implemented the following template-based methods.

Layout Averaging (LA): Simply learns the expected spatial template $\hat{s}_t$ for triplet class $t = (S, R, O)$ as the averaged spatial layout across all instances of triplet $t$ present in the training samples. That is, $\hat{s}_t = \frac{1}{N_t} \sum_{i=1}^{N_t} s_i^t$ with $s_i^t$ being the layout of the $i$-th instance of triplet $t$ and $N_t$ the number of training instances of triplet class $t$.

As [69] points out in “The Need for Biases in Learning Generalizations”, biases can be useful and necessary elements in machine learning. The physical world is characterized by natural biases. Spatial templates and spatial clues are essentially biases which can be powerful predictors for scene understanding. In this paper, we study the optimal way of representing such biases (i.e., spatial templates) and demonstrate their effectiveness in a real-world task (HOI detection).

2) LAYOUT-CLASSIFICATION METHODS

Given an observed spatial layout $s$ as input, the model $f()$ outputs scores $f(s) = (f_1(s), \cdots, f_T(s))$ with $f_i(s) = \tilde{P}(T = t_i | s)$ indicating the confidence that the observed layout $s$ belongs to triplet class $t_i$; where $T$ is a random variable and $T$ is the set of triplet classes. The layout-class match $m(t, s)$ between the observed layout $s$ and a given candidate triplet class $t$ is taken as the confidence estimate (Fig. 5, bottom):

$$m(t, s) = f_i(s) \quad (11)$$

We implemented the following layout-classification methods.\,\footnote{With $\tilde{P}(T = t_i | s)$ we abuse notation for the sake of clarity. In practice, the confidence scores $f_i(s)$ do not need to define a probability distribution (i.e., adding up to 1), nor to be between 0 and 1.}

Random Forest (RF) [70]: Is a classical ensemble of decision trees, known to perform strongly in a wide variety of classification problems [71]. Additionally, RF tend to produce well-calibrated probability estimates [72].

Neural Network Classifier (NN): We use a simple feed-forward neural network as a classifier in a layout-classification setting described above.

$$\hat{t} = W_{out} \Re L U(W_h s + b_h) + b_{out} \quad (12)$$

where $\hat{t}$ is the predicted triplet class, $s$ is the observed spatial layout, and $W_h, W_{out}$ are the hidden and output weight matrices respectively, and similarly for the biases $b_h, b_{out}$. The model minimizes the cross-entropy loss $L(t, \hat{s}_t)$ where $t$ is the ground truth triplet class for the layout $s$.

Linear Discriminant Analysis (LDA): LDA can be thought of a more refined version of the layout averaging (LA) method, encoding uncertainty. Instead of solely learning the parameters $\Theta_t$ of the averaged spatial layout for each triplet $t$ (i.e., $\Theta_t = [c_s^T, c_r^T, h^S, w^S, c_o^T, o^O, h^O, w^O]$), LDA learns the probability distributions of such parameters – assumed to be Gaussian – through an additional set of parameters of variances and co-variances, $\Sigma_t$. LDA assumes that data for each triplet class $t$ are normally distributed where $P(S|T = t_k)$ equals to:

$$\frac{1}{(2\pi)^{d/2} |\Sigma_k|^{1/2}} \exp\left(-\frac{1}{2} (S - \mu_k)^T \Sigma_k^{-1} (S - \mu_k) \right) \quad (13)$$

where $S$ and $T$ are the random variables spatial layout and triplet class respectively, and $d$ the number of features (i.e., the dimensionality of $S$). LDA assumes that the covariance matrix $\Sigma$ is the same across classes, i.e., $\Sigma_1 = \cdots = \Sigma_3$. A classifier that predicts posteriors $P(T = t_k | S)$ can be built by Bayes rule:

$$P(T = t_k | S) = \frac{P(S|T = t_k) P(T = t_k)}{P(S)} = \frac{P(S|T = t_k) P(T = t_k)}{\sum_j P(S|T = t_j) P(T = t_j)} \quad (14)$$

where the $P(T = t_k)$ are the (triplet) class priors, which are estimated as the class proportions in the training data.

Quadratic Discriminant Analysis (QDA): Is a more flexible version of LDA, which allows the covariance matrices $\Sigma_1, \cdots, \Sigma_3$ to differ across triplet classes $t_1, \cdots, t_3$.

K-Nearest Neighbor (kNN): We employ the probabilistic predictions of a k-nearest neighbor classifier as category confidence scores.

C. SPATIAL LAYOUTS CONSIDERED

In our tasks, we consider the following spatial layouts $\phi(s_t)$ induced by the composition of different transformations $\phi$ (Sect. IV-A1). Spatial templates are learned from these layouts with the methods above (Sect. IV-B).

Absolute:

- **Absolute (Abs) coordinates:** $s_t = [b^S; b^O]$ with $b^S = [c_s^T, c_r^T, h^S, w^S]$. Idem for $b^O$.

- **Absolute + mirror (Abs + Mirr):** The same as Abs yet mirrored when necessary (Sect. IV-A1.b), i.e., $b^S = [M(c_s^T, c_r^T, h^S, w^S)]$. Idem for $b^O$.

- **Corners** of the bounding box: $s_t = [b^S; b^O]$ with $b^S = [s^S_{\text{left}}, s^S_{\text{top}}, s^S_{\text{right}}, s^S_{\text{bottom}}]$. Idem for $b^O$. This is perhaps the most common way of encoding bounding boxes (e.g., [58] uses it for image generation).

Relative:

- **R-CNN-rel:** As an external baseline that uses relative coordinates we implement the same parametrization used by R-CNN [73]. We consider $S$ as their
The detected objects. Reference [78] encodes objects with the size and position of their bounding box for tracking Transformers where they also need to encode the location relative and absolute coordinates, while [45] is a relative one. [43]'s spatial representation described above contains both detection [28], [73] and HOI detection [8], [43]. For instance, text-based image retrieval [59] and referring expressions such as image captioning [61], image generation [58], [59], Absolute in the literature, and trends appear to be task-related. Means of encoding object locations (boxes) are popular in the literature, and trends appear to be task-related. Absolute coordinates $b = [cx, cy, h, w]$ are popular in tasks such as image captioning [61], image generation [58], [59], text-based image retrieval [59] and referring expressions (a.k.a. visual grounding) [74]–[76]. [75], [76] further add the product $h \ast w$ to the features. Relative coordinates (w.r.t. other objects or anchors) are commonly used in object detection [28], [73] and HOI detection [8], [43]. For instance, [43]'s spatial representation described above contains both relative and absolute coordinates, while [45] is a relative one. Recently, we have also seen an increase in the use of Vision Transformers where they also need to encode the location of bounding boxes. For instance, [77] encodes objects with the size and position of their bounding box for tracking the detected objects. Reference [78] encodes objects with $(x_{LT}, y_{LT}, x_{RB}, y_{RB})$ where $(x_{LT}, y_{LT})$ and $(x_{RB}, y_{RB})$ denote the coordinate of the top-left and bottom-right corner respectively. Where $W$ and $H$ are the width and height of the image respectively. Afterwards, this 4-d representation is embedded into high-dimensional representation by computing sine and cosine functions of different wavelengths. Reference [79] encode the spatial location by constructing a 5-d vector from the normalized top-left and bottom-right coordinates together with the fraction of image area covered. This is then projected to match the dimension of the visual feature and they are summed.

The spatial features ([43], [45]) are compared to our canonical layout in our experiments (Tab. 2).

### V. TASKS

The triplet categorization task (Sect. VII-A) is a laboratory experiment specifically designed to determine the predictive power of spatial information alone (RQ 1). However, methodological transparency demands keeping the design as clean and minimalist as possible by removing any potentially confounding elements. By using ground truth boxes at test time instead of automatically generated object box proposals, we avoid: (1) seeing pixel values at any stage, (2) masking the predictive power of spatial clues with auxiliary elements such as the box proposal network or extra algorithms to rank the predictions of each pair of proposed $S$ and $O$ boxes (Fig. 7), which are by themselves performance bottlenecks. Since availability of human-annotated object boxes at test time is unrealistic in real-world applications, HOI detection (Sect. VII-B) provides further insight on RQ 1 in a real-world setting. However, since this task uses automatic object proposals, neither full exclusion of pixel values (and of proposed object categories), nor independence of auxiliary elements can be claimed.

HOI detection has an explicit spatial character as its performance measure directly judges spatial ability (i.e., localizing the objects), whereas categorization evaluates the model’s spatial capability in an indirect manner. Hence, HOI detection is especially meant to study the effectiveness of different spatial representations (RQ 2).

### A. TRIPLET CATEGORIZATION (LABORATORY)

A natural and interpretable task to study the predictive power of spatial information in scene understanding (RQ 1) is to classify $(S, R, O)$ triplets by using exclusively bounding box coordinates as input, and nothing else (Fig. 6).

1) **CLASSIFICATION PROBLEM FORMULATION**

Formally, we learn a classifier $p : [0, 1]^8 \rightarrow [0, 1]^T$ that predicts the triplet class $t = (S, R, O) \in T$ (output) from a given spatial layout $s' \in [0, 1]^8$ (input), where $s'$ is the i-th layout instance and $t \in T$ its associated triplet class. The data for each classification problem consists of $X = \{s'_i\}_{i=1}^N$ inputs and $y = \{t'_i\}_{i=1}^N$ outputs, where $N = \sum_{t \in T} N_t$, with $N_t$ the number of samples that belong to class $t$. 

![FIGURE 6. Laboratory Triplet categorization task. The red bounding box indicates the object $O$ while the blue bounding box indicates the subject $S$.](image-url)
Notice the challenge of this task. Although no pixel values are given to the purely spatial methods (which would enable recognizing the objects’ class), the classifier still needs to predict the class of both objects and their relationship, to classify \((S, R, O)\).

2) MULTI-CLASS SCENARIO
It is instructive to find the attainable (purely spatial) performance at different degrees of challenge (Fig. 8). We consider learning problems with varying number of triplet classes \((S, R, O)\), randomly drawn from the 600 most frequent classes \(t\) in VG (description of the dataset: see below).

\(^7\)We do not consider the full VG dataset since, unlike HICO-DET \(|\mathcal{T}| = 600\), VG has a non-curated and extremely large number of triplet classes \(|\mathcal{T}| > 400K\), most of which have a single sample, being thus impossible to use at least one sample to train and another to test.

\(^8\)Notice that in HOI detection spatial methods make use of the object categories proposed by R-CNN. Otherwise one would not only need to consider \((n^f)^2\) \(S\)-\(O\) proposals at each image (instead of \(n^f_O\) proposed \(S\)-\(O\) box pairs.

\(\mathbb{a}:\) CLASSIFIER
This task aims to find a good decoder, ideally, one that performs as close as possible to Bayes accuracy (i.e., maximum attainable accuracy). We tested multiple classifiers including: random forest (RF) \([70]\), LDA, \([80]\), feed-forward nets (NNs), and k-nearest neighbor (kNN). Although RF is a generalist classifier known to perform strongly often without hyperparameter tuning \([71]\), \([81]\), other classifiers such as NNs can be very effective in specific conditions, as discussed in Sect. VII-A3. Please note that the classifier is also trained to predict the ‘no_relation’ relationship. (and leave out the purple stuff) The ‘no_relation’ relationship is treated like the rest of relationships for training and testing.

\(\mathbb{b}:\) BOUNDING BOX SOURCES
As argued above, ground truth box coordinates (without object class labels) are employed at both train and test time for methodological transparency.

B. HOI DETECTION (REAL-WORLD)
The benchmark task of HOI detection \([8]\) consists in predicting both the triplet classes \(t = (S, R, O)\) as well as their associated \(S\) and \(O\) bounding boxes, given an image as input. The dataset for this task, HICO-DET \([8]\) (description of the dataset: see below), consists of instances \(t = (S, R, O)\) of humans interacting with objects, where the Subject is always a person (Sect. VI-A).

1) DETECTION PROCEDURE
Training: Learn either a Template-based or a Layout-classification model (Sect. IV-B), using the ground truth annotated boxes on HICO-DET.

Test:
- **Box proposal.** Given an image \(I\), generate box proposals with their predicted object class using Faster-RCNN \([28]\) trained on COCO \([82]\). This yields \(n^f_S\) Subject person) and \(n^f_O\) Object box proposals. \(n^f_S\) contains exclusively all detected “person”, and \(n^f_O\) contains all detected “person” (since a person can also be an Object) plus all detected objects of any class that belongs to our Object vocabulary \(O = \{O | \exists (S, R, O) \in \mathcal{T}\}\).
- **S-O box proposals.** Next, each proposed \(S\) box is combined with each proposed \(O\) box, yielding \(n^f_S \times n^f_O\) proposed \(S\)-\(O\) box pairs.
- **Inference (prediction).** For each proposed \(S\)-\(O\) pair \(j = 1, \ldots, n^f_S \times n^f_O\) in image \(I\), compute its match \(m(s_j, t)\) against each of the triplet classes \(t \in \mathcal{T}_O\) that are possible \(^9\) for the Object class \(O\) of proposal \(S\)-\(O\), i.e., \(\mathcal{T}_O := \{(S, R, O) \in \mathcal{T} | O = O\}\). For example, if \(O\) is “bike”, then (person, riding, bike) and (person, walking, bike) are possible triplets, but (person, riding, horse)
is not. The class $O^i$ may be either predicted or given (Sect. VII-B1). The match is computed according to eq. 10 and 11. At each $S$-$O^i$ pair, this yields an output of $|T_{O^i}|$ tuples of the form: $(m(s_j, t), b_{ij}^S, b_{ij}^O)$ with $t \in T_{O^i}$. Notice that $b_{ij}^S, b_{ij}^O$ are the same for each $t \in T_{O^i}$.

VI. DATASETS AND IMPLEMENTATION

A. DATASETS

The two datasets of our experiments are described below.

Visual Genome (VG): The VG dataset [68] contains 108,077 images that are human-annotated with $\sim$1.5M triplet ($S, R, O$) instances. Each instance includes bounding boxes for both $S$ and $O$. In total, there are $\sim$25k unique $Rs$, $\sim$39k unique $S$s, and $\sim$33k unique $Os$. On average, each image has 35 objects and 21 pair-wise relationships between these objects. We use version 1.2 of VG.9 We also wish to acknowledge the existence of the Visual Genome HOI dataset [56]. However, we found it less popular than HICO-DET [8]. Additionally, in the setting where we use VG, we wish to complement HOIs with other triplets where the Subject is not a person and where the relation is not an action. Hence, we do not use this dataset in the paper.

HICO-DET: The Humans Interacting with Common Objects - Detection or HICO-DET [8] is a dataset created for detecting Human-Object Interactions (HOI) and is built on the HICO dataset [40]. It consists of 47,776 images that depict humans (i.e. $S = 1$) performing 117 different type of actions ($R$). In total there are 600 different interactions with 80 different object classes ($O$), which correspond to the 80 classes of the COCO dataset [82]. Over all the images there are 90,641 HOI instances and 256,672 bounding boxes that indicate the location of the 80 object classes and humans in all the images. The training set split consists of 70,373 HOI instances and the test set of 20,268.

VG and HICO-DET exhibit conceptual differences that make them representative of a wider variety of problems. First, in HICO-DET all Relations are implicitly spatial (i.e., actions and thus there are no “left of” or “right of” relations) while in VG a considerable proportion of triplets are explicitly spatial (e.g., $R = \text{“on”}$ is present in 38.4% of the relationships, “above” 0.66%, “left” 0.09%, “right” 0.08% etc.). Additionally, while VG has a wide variety of possible $S$s (different objects), $S$ in HICO-DET is always a person.

Please note that these datasets do not contain depth information. In the case of HICO-DET, there are also no relationships like “in front” or “behind”.

Validation Setting: Because VG [68] does not provide any train / test splits, we use 3-fold cross-validation in triplet categorization.10 That is, data are randomly split into three stratified folds (by triplet class), and 2/3-rds are used for training and 1/3-rd for testing. Reported results are averages and standard deviations (SDs) over the three folds, i.e., using a different fold as a test set each time. The choice of 3-fold cross-validation is motivated by the large size of the VG data and the stability of the results (see e.g., the almost negligible variances across folds in Tab. 1 (Supplement) or Fig. 8, generally too small to be seen). In HICO-DET, reported results are averages and SDs over three runs of the models with different seeds using the given train/test splits.

B. IMPLEMENTATION AND HYPERPARAMETER SETTING

For specific and technical details about hyperparameter and implementation settings, see the Appendix.

---

9https://visualgenome.org

10Only HICO-DET (and not VG or VG-HOI [56]) is used in HOI detection for several reasons. First, HOI detection needs the Subject to always be a human – and this is not the case in VG. Second, the object classes of our pre-trained object detector (Faster-RCNN) are the same as in HICO-DET but not in VG. Third, there is an overlap of images between COCO (where Faster-RCNN is trained) and VG, hence rendering our current cross-validation setting (Sect. VI-A) not feasible as this would imply testing in images (in VG) that were used to train Faster-RCNN (in COCO). Finally, external baselines exist only for HICO-DET [8].
VII. EXPERIMENTS AND RESULTS

A. TRIPLET CATEGORIZATION (LABORATORY)

1) EXPERIMENTAL SETUP

**Evaluation Measures:** For interpretability, we report **accuracy** (a.k.a. correct classification rate). To control the class imbalance between triplet classes, we also compute: averaged class recalls (a.k.a. macro accuracy), averaged precision, and F1 measure averaged across classes. All measures show similar performance patterns (see Supplement).

2) BASELINES AND COMBINED METHODS

The following methods are changes or combinations of the inputs to the classifier.

**Visual Appearance (CNN):** To measure the strength of only using spatial information, we need to compare with a model that uses visual features. To obtain visual representations, we extract 2,048-dimensional features from the last average-pooling layer with the forward pass of the ResNet-50 CNN model [83] trained on ImageNet following [47]. We extract CNN features from the *crop of the tightest box* (TB) that contains both S and O boxes, where \( b^S, b^O \) are ground truth boxes. In both tasks, CNN features are employed in the same manner as spatial layouts \( s \). That is, 8-dimensional layout vectors \( s \) are replaced by a 2,048-d CNN vectors \( v \). Crucially, notice that the CNN uses the same ground truth annotated boxes as our spatial methods (Sect. VII-A), yielding, therefore, a fair comparison.

**Early Fusion (CNN+CPN+MirrEarly):** The CNN features \( v \) above are concatenated \( ([x; v]) \) with the transformed spatial layouts \( s \) and fed as input to a random forest\(^{11} \) classifier.

**Rand (Chance Baseline):** Spatial layouts \( s \) are randomly drawn from a uniform \([0, 1]\) distribution, dimension-wise.

3) RESULTS

**Notation:** The different spatial layouts are denoted as in Sect. IV-C, and visual and fused methods as above (Sect. VII-A2). For clarity, methods that are not essential to understand the results landscape are left to the Supplement.

It is first worth noting the remarkable predictive power of spatial clues *alone* in a *semantic* task (Fig. 8). Given 50 different triplet types \((S, R, O)\), around 50-65% of the scenes can be predicted correctly from their spatial layouts alone. Notice that the baseline that uses random feature vectors (Rand) attains higher accuracy (9-12%) than chance (2%) as it learns to predict constantly the category with the most samples. The exact numbers of Fig. 8 are reported in the first table of the Supplement.

\(^{11}\) A multi-layer perceptron provided similar results as a classifier. Late fusion was additionally considered, which fared slightly worse than early fusion with either type of classifier.

\[ b: SPATIAL PLUS VISUAL \]

In Fig. 8 we generally observe a performance gain (\( \sim 1-8 \) accuracy points) by fusing spatial and CNN features (CNN+CPN+MirrEarly), which indicates a non-redundancy of visual appearance and spatial clues, further suggesting great potential in explicitly modeling both types of information in a system. In HICO-DET the gain is less outspoken but note that because of their architecture CNN features already encode some spatial information.

\[ c: BINARY CLASS \]

Given the interpretability of this task and of its performance measure (accuracy), we manually select pairs of triplet classes \((S, R, O)\) for *illustrative purposes* (Tab. 1). It is first worth noting the generally high accuracy (often above 90%) by which two different scene classes can be recognized by using exclusively the spatial locations and sizes of the objects. As expected, very similar scene pairs are harder to distinguish, e.g., (player, holding, hat) versus (player, swinging, bat), than very different ones, e.g., (person, flying, kite) versus (person, riding, horse). Additionally, some scene pairs are easier to distinguish from their visual appearance than from their spatial layouts, e.g., (man, holding, racket) versus (man, holding, camera), especially those involving different objects and the same action \((R)\). However, some scene pairs may be easier to distinguish from their spatial layouts than from visual appearance, e.g., (person, riding, horse) versus (person, walking, horse), as the two scenes share the same object type ("horse") while the spatial layout of the two actions ("walking" vs. "riding") substantially differs.

\[ d: LAYOUT TRANSFORMATIONS \]

There seems to be no clear advantage of one spatial representation over the others (Fig. 8), none proves consistently superior across datasets. Because a classifier may learn by itself to perform the appropriate transformations to the input (box coordinates) that are most helpful in predicting the class \( t = (S, R, O) \), this task rather provides insight on **RQ 1**. That is, how far can we get in predicting the class \((S, R, O)\) by using exclusively spatial information.

\[ e: CLASSIFIER CHOICE \]

Fig. 8 shows that NNs are especially effective in both high-dimensional (CNN features) and large data problems when considering a large number of triplet classes (both around 55% accuracy). In the HICO-DET dataset, CNN performs better. However, the Abs model still attains a good accuracy even for a large number of triplet classes where the difference in performance compared to CNN decreases. The salience of this result lies in that this is a purely *semantic* task, rather than **spatial**. It is also worth noting the drastic compression of spatial layouts \((8-d)\) compared to visual features \((2,048-d)\). Overall, this result indicates that, oftentimes, spatial clues alone may provide a predictive power comparable to visual clues for scene understanding.
Following [8], we consider two experimental conditions (e.g., 600 triplets). In contrast, RF fares better with low-dimensional features (i.e., spatial). This is generally the expected behavior, as high-dimensional data with many dependencies may pose a challenge to RF [84], and NNs are known to deal well with high-dimensional data by learning latent features. The relative performance of the methods (i.e., their ranking) is preserved across classifier choices, hence the conclusions above remain similar.

### B. HOI DETECTION (REAL-WORLD)

#### 1) EXPERIMENTAL CONDITIONS

Following [8], we consider two experimental conditions:

- **Known Object**: the classes $O_t$ of all ground truth Objects present in the image $I$ are given.\(^\text{12}\) Hence, one needs to predict all instances of the triplet classes $t = (S, R, O')$ ($\forall O' \in O_t$) present in the image, as well as their corresponding $b^S$ and $b^O$ boxes. Since $O'$ is given and $S$ is always a person in HICO, predicting $t = (S, R, O')$ amounts to predicting only $R$.

- **Default** setting: Is more challenging than the “Known Object” condition. Only the image is provided, and no information about the Object classes is given. One must predict all $t = (S, R, O)$ instances present in the image, as well as their corresponding $b^S$ and $b^O$ boxes. Sect. VI-B1 describes this condition.

#### 2) EXPERIMENTAL SETUP

##### 2a: BOUNDING BOX SOURCES

Following [8], we use annotated boxes of HICO-DET for training and box proposals generated with Faster-RCNN [28] at test time. Notice that knowledge of what box proposal corresponds to $O$ and what box to $S$ is also available at test time since each box proposal outputted by Faster-RCNN is accompanied by its corresponding predicted class (e.g., “person” or “bike”), and Subjects are always “person” in HICO-DET. Hence, our layout transformations (Sect. IV-A1) can be applied.

##### 2b: EVALUATION MEASURES

Performance is evaluated with the HOI-mAP proposed by [8], which is a more demanding version of the classical mAP used in object detection [85], [86]. In contrast with the latter, HOI-mAP considers $S$-$O$ box pairs instead of a single object and requires both $S$ and $O$ to be correctly detected. In object detection, a true positive is assigned if the Intersection-over-Union (IoU) overlap between the predicted and the ground truth object box is larger than 0.5. In HOI detection, both Subject $IoU_S$ and Object $IoU_O$ must be larger than 0.5, that is, $\min(IoU_S, IoU_O) > 0.5$. Then, the average precision (AP) per class is computed according to the ranking induced by the matches $m(t, s)$ between observed spatial layouts $s$ and candidate triplet classes $t$ (Sect. IV-B, eq. 10 and 11). Following [8] and the 2007 Pascal VOC challenge [86], the AP per class is computed with the 11-point interpolation, which approximates the area under the precision-recall curve by averaging precision at 11 equally-spaced recall points ($[0, 0.1, \ldots, 1]$). The mean average precision (mAP) is the mean of APs across classes.

Following standard practices [8], [51], we report mAP on three subsets of HOI categories: (1) the full set of 600 HOI categories; (2) the 138 rare HOI categories with less than 10 training samples; and (3) the 462 non-rare HOI categories with more than 10 train samples.

#### 3) BASELINES AND COMBINED METHODS

##### 3a: VISUAL APPEARANCE (CNN\(_{LA}\))

As in triplet categorization (Sect. VII-A) CNN features $v_j$ are obtained from crop of the tightest box that contains both, $b^S$ and $b^O$ boxes of the $j$-th $S$-$O$ pair, where $b^S, b^O$ are now R-CNN proposals at test time and ground truth boxes at train time. Next to using ResNet [83] features, we additionally included bottom-up and top-down (2,048-d) visual features [87] in the present task. We also consider a model with three visual branches (CNN\(^3\)) $v = [v^{TB}, v^S, v^O] \in \mathbb{R}^{6144}$, which extracts a visual vector for the tightest box $v^{TB}$, one for the Subj $v^S$ and one for the Obj $v^O$. CNN features

\(^{12}\)A few images in HICO-DET have more than one $O$ class — the majority of images have a unique object class. There are often multiple instances of the same object class (e.g., bikes) and/or multiple triplet classes $t$ with the same $O$ involved (e.g., (human, repairing, bike) and (human, holding, bike)). There may also be multiple instances of the same triplet class $t$ in the same image.
TABLE 2. HOI detection in the full HICO-DET data. Results are HOI-mAP – the higher, the better. The first and second blocks are our methods and own implementations of external methods, and the third block is external baselines. In the first block, we use the ResNet-101 Faster R-CNN implementation from [88] while in the second block, we use the ResNet-152 Faster R-CNN implementation from [89]. Subscripts indicate which learning method from Sect. IV-B is employed (e.g., Reg or QDA) – for brevity we do not any subscript when the method uses layout averaging (LA). For instance, CPN + Mirr corresponds to CPN + Mirr LA. Notation for fused and visual models (CNN) is as specified in Sect. VII-B. The best method per block (column-wise) is boldfaced, and the best overall is underlined. Standard deviations (SDs) are mostly negligible, hence omitted for readability and left to the Supplement. Columns correspond to the setting from Sect. VII-B1 and VII-B2.

| Method | Backbone | Detector | Feature Extractor | Full | Non-Full | Full | Non-Full | Known Object |
|--------|----------|----------|-------------------|------|----------|------|----------|---------------|
| Rad-vn | ResNet-101 [83] | Faster R-CNN [28] | None | 1.97 | 2.08 | 1.94 | 1.97 | 3.85 | 3.45 | 3.99 | 3.06 | 5.66 | 6.2 |
| RCNN-scores [88] | ResNet-101 [83] | Faster R-CNN [28] | None | 3.85 | 3.35 | 3.99 | 6.08 | 5.66 | 6.2 |
| RCNN-rel | ResNet-101 [83] | Faster R-CNN [28] | None | 4.53 | 2.49 | 5.13 | 8.05 | 5.13 | 8.92 |
| Corners | ResNet-101 [83] | Faster R-CNN [28] | None | 7.17 | 6.2 | 7.47 | 10.31 | 9.04 | 10.72 |
| Abs | ResNet-101 [83] | Faster R-CNN [28] | None | 8.08 | 7.27 | 8.32 | 11.46 | 10.41 | 11.77 |
| Abs+Mirr | ResNet-101 [83] | Faster R-CNN [28] | None | 8.15 | 7.6 | 8.32 | 11.44 | 10.4 | 11.75 |
| Cr+Mirr | ResNet-101 [83] | Faster R-CNN [28] | None | 8.44 | 7.64 | 7.66 | 11.98 | 11.11 | 12.25 |
| CPN | ResNet-101 [83] | Faster R-CNN [28] | None | 8.46 | 7.42 | 8.77 | 12.38 | 10.42 | 12.97 |
| CPN+Mirr | ResNet-101 [83] | Faster R-CNN [28] | None | 8.82 | 7.88 | 9.1 | 12.89 | 11.43 | 13.32 |
| CPN+Mirr+DA | ResNet-101 [83] | Faster R-CNN [28] | None | 6.64 | 5.8 | 6.9 | 10.45 | 9.12 | 10.84 |
| CPN+Mirr_LA | ResNet-101 [83] | Faster R-CNN [28] | None | 4.37 | 3.97 | 4.83 | 7.35 | 6.68 | 7.55 |
| CPN+Mirr+LA | ResNet-101 [83] | Faster R-CNN [28] | None | 5.65 | 3.35 | 6.34 | 9.02 | 6.99 | 9.63 |
| CPN+Mirr+LA | ResNet-101 [83] | Faster R-CNN [28] | None | 4.84 | 4.37 | 4.99 | 7.46 | 6.84 | 7.64 |
| CPN+Mirr+LA | ResNet-101 [83] | Faster R-CNN [28] | None | 7.63 | 6.86 | 7.96 | 11.32 | 9.32 | 11.92 |
| CPN+Mirr+LA | ResNet-101 [83] | Faster R-CNN [28] | None | 6.92 | 5.84 | 6.86 | 9.14 | 9.92 | 8.9 |
| CNN+CPN+Mirr_LA | ResNet-101 [83] | Faster R-CNN [28] | ResNet-50 [83] | 10.85 | 10.03 | 11.1 | 13.78 | 13.25 | 13.94 |

v ∈ R^{2048} are used in an identical fashion as the spatial layout vectors s ∈ R^8 at each S-O proposal, at both learning (layout averaging) and inference time.

\[ b: \text{LATE FUSION (CNN+CPN+Mirr\_Late)} \]

The joint match \( m_{\text{late}}(t, s) \) of the visual (in the previous paragraph) and the spatial model (Sect. V-B1) is fused at each S-O proposal as:

\[ m_{\text{late}} = \alpha \cdot m_{\text{spat}} + (1 - \alpha) \cdot m_{\text{vis}} \]

where \( \alpha \in [0, 1] \). In this paper we set \( \alpha = 0.5 \).

c: RCNN-SCORES

Predicts the triplet classes \( t \) (output) by using the confidence scores from Faster-RCNN as input. For each proposed S-O box pair, the input is \([r^S_i, r^O_i]\), with \( r^S_i \) the Faster-RCNN confidence scores for the \( S^i \) and the \( O^i \) box, respectively.

\[ d: \text{Rand-m} \]

The match \( m_{\text{rand}}(s, t) \) at each S-O proposal is drawn from a random uniform [0, 1], and Faster-RCNN proposed boxes are used.

e: MODELS PROPOSED

The main idea proposed by the paper is using exclusively spatial clues in HOI detection and triplet categorization. Thus our contribution encompasses all spatial-only methods from Table 2 (e.g., Abs, etc.). However, we put the spotlight on the

\[ Chao et al. [8] find RCNN-scores to be their strongest baseline, excluding their main model. We deemed it more appropriate to report [8]'s performance directly from their paper given that their main model has multiple hyperparameters to tune, has an involved training procedure and employs a different framework (Matlab) than ours (Python). Hence, implementing it ourselves was unlikely to yield a fair representation of their performance. Additionally, our train-test splits of HICO-DET are the same as in [8], and hence comparable.

\[ e: MODELS PROPOSED \]

The main idea proposed by the paper is using exclusively spatial clues in HOI detection and triplet categorization. Thus our contribution encompasses all spatial-only methods from Table 2 (e.g., Abs, etc.). However, we put the spotlight on the
canonical template, i.e., CPN+mirror (+IoU), rather than on Abs, etc. Another key point of this paper is the combination of visual and spatial clues, hence we also emphasize our proposed (best-performing) CNN³ +CPN+Mirr+IoULate.

4) RESULTS

HOI detection is a scene understanding task that is particularly informative about a model’s spatial ability.

a: LAYOUT TRANSFORMATIONS

A systematic analysis of Tab. 2 reveals that:

- **Effect of mirroring**: The Mirr transform (Abs+Mirr and CPN+Mirr) seems to improve the non-mirrored counterparts (Abs and CPN) only slightly.
- **Effect of translation invariance**: Using relative coordinates (CPN and Ctr), i.e., effectively making the layouts translation-invariant, generally improves performance w.r.t. absolute coordinates (Abs) relatively by around 5% in the Full category of the Default condition and 7% for the Full category of Known Object. This is not the case for RCNN-rel, arguably because of the information loss in the compression from 8-dimensional representations (Abs) to 4-d (RCNN-rel) (Sect. IV-C).
- **Effect of scale invariance**: Methods that are both, translation and scale-invariant (CPN+Mirr) outperform those that are only translation-invariant (Ctr+Mirr) relatively by 4.5% in Full category of the Default condition and 7.5% in the Full category of Known Object.
- **Adding the IoU as an extra spatial feature 2)** yields improved performance over the same model without this feature i.e., CNN+CPN+Mirr+IoULate has a 3.5% relative improvement over CNN+CPN+Mirr in the Rare category of the Default condition and 2.9% in the Rare category of Known Object.

Overall, CPN+Mirr outperforms the rest of the spatial methods, including other relative coordinates (RCNN-rel, Ctr+Mirr or Peyre et al. [45]). This supports the use of canonical templates as a fast predictor for tasks where spatial cues are relevant.

b: LEARNING METHODS

Notably, the simple layout averaging method (LA) provides the best results (CPN+Mirr and CNN+CPN+MirrLate), suggesting that template-based methods (Sect. IV-B1) are more effective than the layout-classification methods such as CPN+MIRR_LDA, CPN+MIRR_QDA, CPN+MIRR_RF, etc. (Sect. IV-B2) relatively by 25%-148% in the Full category of the Default condition and 9%-88% in the Full category of Known object when comparing the classification methods with CNN+CPN+MirrLate.

c: SPATIAL VERSUS VISUAL

The canonical spatial template CPN+Mirr with ResNet-152 features outperforms visual appearance (CNN), except the 3-branch version (CNN³) of CNN features, by 3-29% relatively for the Full category in Default and 11-36% for Full in Known Object.

d: SPATIAL PLUS VISUAL

We notice a clear gain by fusing CNN and spatial methods (CNN+CPN+MirrLate) w.r.t. each method alone, especially in the most challenging Default condition. This suggests the non-redundancy of visual (semantic) and spatial clues in images, which is particularly important for tasks that exhibit both, a semantic and a spatial component such as HOI detection.

e: COMPLEMENTARY EVALUATION MEASURES

Even though mAP is the ubiquitously adopted and standard performance measure in HOI detection, given the sparsity of annotations in our datasets we estimated relevant to also include recall. Tab. 3 shows results with recall@50 and recall@100 using the evaluation script’s implementation of [32]. Results show that using only spatial clues suffices to attain remarkably high results, comparable to the best-performing state-of-the-art method, [45], in terms of mAP. The canonical template shows lower recall@k than absolute coordinates (Abs). We hypothesize that because the canonical template implements multiple types of invariance, it is less able to retrieve particular cases of individual spatial layouts – as a result of having collapsed many potential biases present in the images in order to achieve invariance. The reader can find results with (macro-averaged) recall in the Supplement (Tab. 2) for the triplet categorization task.

| Method                        | Recall@50 | Recall@100 |
|-------------------------------|-----------|------------|
| Abs                           | 44.05     | 48.75      |
| CPN+Mirr+IoU                  | 38.96     | 43.69      |
| CNN³+CPN+Mirr+IoULate         | 51.26     | 53.34      |
| Peyre et al. [45]             | 49.66     | 54.05      |

f: EXTERNAL BASELINES

One can expect that the best HOI classifiers are those baselines optimized for performance in an end-to-end fashion, perform hyperparameter tuning, and combine multiple knowledge streams. Thus, it is worth noting that the simple layout-averaging (LA) does not tune any hyperparameters, this training time is close to zero, it is non-end-to-end, and it combines only one or two knowledge streams (spatial and/or visual). Hence, its performance is expected to be sub-optimal so having ample room for improvement. Nevertheless, LA methods (e.g., CNN+CPN+MirrLate and CPN+MirrLA) often perform competitively with SOTA baselines (Tab. 2), especially in the ‘rare’ subset where they often compare favorably i.e. CNN+CPN+MirrLate outperform No Frills [43] relatively by 3% in the Default condition. Also note that CNN³+CPN+Mirr+IoULate outperforms all methods except PMFNet [46] and [45] in the Rare category in both Default as Known Object. It is also notable
that simple non-neural-based methods can outperform some deep end-to-end architectures (e.g., Gao et al. [51], Gkioxari et al. [52]).

It is also important to notice that external baselines (Tab. 2) generally use external knowledge and model components learned elsewhere, which makes it difficult to compare their performance fairly based solely on each method’s merit. For instance, [43], [46], [49] use a pose estimator (i.e. OpenPose [94] or CPN [95]) trained on an external dataset (i.e. COCO [96]). [42], [45], [49] employ pre-trained word embeddings (i.e. Word2Vec [97]/FastText [98]) and Xu et al. [49] uses <object1, predicate, object2> triplets from the “visual relationships dataset” [32].

g: RARE vs. NON-RARE
Tab. 2 shows that simple Layout Averaging (LA) methods exhibit the smallest performance drop from the rare to the non-rare subset. In contrast, external baselines show notable performance drops in their rare compared to the non-rare subset. Since LA learns templates simply by averaging across training samples, it is insensitive to class-imbalance by design, as there is no global loss that drives learning. In contrast, external baselines are often end-to-end neural systems, driven by a global loss that may reward ignoring rare categories. We hypothesize that this plays an important role in explaining these results.

h: EXPERIMENT: REDUCING DATA SIZE
To further study the behavior of our template-based method with a small amount of data, we artifically undersample the training set, keeping only 10 instances per triplet category – and leaving untouched the subset of rare classes which already have 10 samples or less. We do the same with two state-of-the-art methods with publicly available code, [42], [45], by re-training them in the under-sampled set. Table 4 shows that neural network-based external baselines [42], [45] incur a drastic performance drop, in contrast with our simple layout-averaging (LA) template-learning strategy which performs remarkably close to the full data regime. It is worth noting the drastic data reduction from 117,870 training samples (S, R, O) originally, to 5,015 in the undersampled set. Finally, we notice that the performance drop of the LA (template-based) method is smaller than that of the NN (layout-classification) when comparing full data (Tab. 4, top) to small data settings (Tab. 4, bottom).

i: IMPLEMENTATION CHOICES
To better understand the performance landscape, it is instructive to evaluate the effect of implementation choices and model components on performance. Tab. 2 highlights the impact of these decisions. Notable gains are obtained with a ResNet-152 implementation [89] of the R-CNN object detector [28], compared to a ResNet-101 implementation [88]. Further improvements can be obtained by using better CNN visual features [87]. Importantly, these results support that the improvements of our method CPN+Mirror (w.r.t. Abs) are robust against implementation choices. A general takeaway is that caution must be taken when comparing the overall performance of any two baselines, since often, implementation choices have a larger impact than the model itself, as pointed out by many [99], [100].

j: EXTERNAL SPATIAL FEATURES
We notice that our canonical layout (CPN+mirror) outperforms existing spatial features such as Peyre et al. [45] (spatial features) and No Frills [43] (spatial features) (Tab. 2), including the 21-dimensional spatial vector from [43] built with hand-crafted features by at least 6% relatively on the Full set of Default and 4.9% on Full from Known Object. This is a key result of this paper as it supports that our canonical layout is a somewhat optimal way of representing spatial layouts due to the multiple types of invariance that are built into it.

k: COMPUTATION TIMES
Here we quantify the computation times at the prediction or inference stage, which has implications on the feasibility of a method for real-time applications. Tab. 5 shows the average inference times per image where the total inference time \( T^{\text{total}} = T^{R-CNN} + T^{\text{match}} \) is composed of (i) the time to generate R-CNN object proposals in image \( I \) (\( T^{R-CNN} \)), plus the time to compute the multiple matches \( m(t, s_j) \) per image (\( T^{\text{match}} \)), i.e., at each S-O proposal pair \( j \). In turn, \( T^{\text{match}} = T^{\text{feat}} + T^{\cos} \) is composed of: (ii) extracting features (\( T^{\text{feat}} \)) i.e., either computing the transformed layout \( s_j \) (spatial methods) or extracting CNN features \( v_j \) (CNN); and finally (iii) computing cosine similarity (\( T^{\cos} \)). There are, on average, \( \sim 410 \) S-O proposals per image

---

### Table 4. Experiment with small amount of data, where the training set has been reduced to 10 samples per triplet category. Methods in this table use the ResNet-152 [85] object detector backbone and BUTD [87] visual features. Although our ‘rare’ subset has roughly as many samples per category as the ‘full’ subset after undersampling, we keep the same subsets for the sake of comparison with Tab. 2.

|                  | Full Data | Rare Data | Non-Rare Data |
|------------------|-----------|-----------|---------------|
| **Fill Data**    |           |           |               |
| Abs              | 9.54      | 8.73      | 9.79          |
| CPN+Mirror+IoU   | 11.02     | 9.62      | 11.44         |
| CPN+Mirror+IoU_NN| 10.18     | 8.26      | 10.76         |
| CNN\(^3\)       | 13.82     | 12.53     | 14.21         |
| CNN\(^3\)+CPN+Mirror+IoU | 15.4      | 14.07     | 15.8          |
| HOID [42]        | 17.85     | 12.85     | 19.34         |
| Peyre et al. [45]| 19.4      | 14.6      | 20.9          |
| **Small Data**   |           |           |               |
| Abs              | 9.55      | 8.73      | 9.79          |
| CPN+Mirror+IoU   | 10.73     | 9.62      | 11.06         |
| CPN+Mirror+IoU_NN| 8.71      | 7.33      | 9.13          |
| CNN\(^3\)       | 13.52     | 12.53     | 13.81         |
| CNN\(^3\)+CPN+Mirror+IoU | 15.1      | 14.07     | 15.4          |
| HOID [42]        | 11.3      | 9.5       | 12            |
| Peyre et al. [45]| 5.03      | 5.2       | 4.98          |
(i.e., feature extraction and cosine computations). Notice that all methods in Tab. 5 use the same type of computations for inference (i.e., cosine similarity) and learning methods (layout averaging), and are thus comparable. We see in Tab. 5 that the extra time taken for CPN computations (CPN+Mirr) is almost negligible compared to not performing such transformation (Abs), and that the spatial methods are over two orders of magnitude faster ($T_{\text{match}}$) than CNN. Overall, the use of spatial clues as compressed and fast scene descriptors is supported.

### C. OVERALL DISCUSSION

Consistently across tasks, all spatial methods performed well-over chance, evidencing that large predictive power can be seized from spatial clues alone (RQ 1) not only in spatially focused tasks (HOI detection), but also in semantic scene understanding (categorization). Additionally, the fusion of spatial and visual (CNN) features improved both spatial and CNN individual methods in all tasks, suggesting that both components are needed for a better scene understanding. Although the canonical spatial representation provided a consistent gain in the task with a spatial emphasis (HOI detection), none of the spatial representations seemed to dominate in the semantic task (triplet categorization), suggesting thus the adequacy of the canonical layout in applications with an accentuated spatial component (RQ 2).

### VIII. CONCLUSION AND FUTURE WORK

Spatial understanding is essential to a plethora of semantically- and spatially-focused scene understanding applications, including image retrieval [15], [62], [101], robot navigation [102], robot understanding of natural language commands [63], [64], object recognition [10], [103] and image generation [58], [104], to name a few. The main contribution is that we empirically show that using exclusively spatial information in triplet categorization and HOI detection provides high performance and is fast in terms of processing time. Overall, our experiments show that spatial clues alone have large predictive power in both semantic and spatial scene understanding tasks. Such a finding is relevant as in time-critical applications operating in realistic open-domain settings spatial cues could filter the many possible objects and actions encountered. We have furthermore shown that by combining visual appearance and spatial clues a performance boost is obtained in detecting objects and actions in a visual scene especially in regimes where few annotated data are available. Overall, this paper focuses on testing fundamental hypotheses involving the role of spatial cues in scene understanding. Hence, we have emphasized methodological transparency over performance optimization. Although straightforward, learning spatial templates as averaged vector prototypes and evaluating semantic match via cosine similarity already provides clear performance gains (and low computational load), which can be taken as a starting point in future research. Also, for methodological clarity, we have used straightforward methods such as early and late fusion of spatial and visual clues. However, we believe improvements can be obtained in future work by using more elaborate methods such as deep gated fusion models and transformer architectures. We also leave as future research the use of neural models capable of predicting unseen object-relationship-object classes (i.e., zero-shot) by leveraging distributed (embedding) representations of objects [13], [45]. It would also be interesting to see how one could recover from the wrong predictions made by the object detector. This is crucial as any error made here gets further propagated down the rest of our pipeline. One could, for instance, use the method proposed by [105] and train multiple object detectors to extract features, bounding boxes, and classes of objects. This information can then be used in the algorithm proposed by [105] to recover from errors and potentially even train a better object detector. Finally, the effectiveness of the proposed canonical spatial layout in a variant of object detection (HOI detection) shows promising potential. Although our work is limited to 2D scene processing, it opens possibilities to evaluate its findings in 3D scene processing when appropriate datasets become available.

Overall, the low computational burden and effectiveness shown by spatial clues pave the way toward using them as low-dimensional scene descriptors to perform fast scene understanding at a glance, in a human-like scene “gist” fashion. Concretely, applications that require fast real-time scene understanding, such as automatic object localization from natural language commands in self-driving cars [12] show particular promise. Furthermore, the strong performance of our methods with small training data sets showcases the potential for problems with a small amount of data. This paper and its results offer an incentive to reflect on novel representations of scenes and its objects that integrate spatial and visual information.

### REFERENCES

[1] G. L. Malcolm, I. I. A. Groen, and C. I. Baker, “Making sense of real-world scenes,” Trends Cognit. Sci., vol. 26, no. 11, pp. 843–856, Nov. 2016.

[2] J. S. Werner and L. M. Chalupa, The New Visual Neurosciences. Cambridge, MA, USA: MIT Press, 2014.

[3] I. Kadar and O. Ben-Shahar, “A perceptual paradigm and psychophysical evidence for hierarchy in scene gist processing,” J. Vis., vol. 12, no. 13, p. 16, 2012.

[4] L. Fei-Fei, A. Iyer, C. Koch, and P. Perona, “What do we perceive in a glance of a real-world scene?" J. Vis., vol. 7, no. 1, p. 10, Jan. 2007.

[5] M. Greene and A. Oliva, “Recognition of natural scenes from global properties: Seeing the forest without representing the trees,” Cognit. Psychol., vol. 58, no. 2, pp. 137–176, Mar. 2009.

[6] Platanov and Schubert [106] use a very simple 3D block and room world to model explicit spatial prepositions such as “on”, “behind”, in” etc.
[6] C.-C. Wu, H.-C. Wang, and M. Pompianu, “The roles of scene gist and spatial dependency among objects in the semantic guidance of attention in real-world scenes,” Vis. Res., vol. 105, pp. 10–20, Dec. 2014.

[7] M. S. Castelhano and C. Heaven, “Scene context influences without scene gist: Eye movements guided by spatial associations in visual search,” Psychonomic Bull. Rev., vol. 18, no. 5, pp. 890–896, Oct. 2011.

[8] Y.-W. Chao, Y. Liu, X. Liu, H. Zeng, and J. Deng, “Learning to detect human-object interactions,” in Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV), Mar. 2018.

[9] E. Zablocki, B. Piwowarski, L. Soulier, and P. Gallinari, “Learning multimodal word representation grounded in visual context,” in Proc. AAAI, 2018.

[10] S.-R. Shiang, S. Rosenthal, A. Gershman, J. G. Carbonell, and J. Oh, “You only look once: Unified, real-time object detection,” CoRR, vol. abs/1506.02640, pp. 1–10, Jun. 2015. [Online]. Available: http://arxiv.org/abs/1506.02640

[11] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, “CenterNet: Keypoint triplets for object detection,” CoRR, vol. abs/1904.08189, pp. 1–10, Apr. 2019.

[12] C. Liu, R. Krishna, M. Bernstein, and L. Fei-Fei, “Visual relationship detection with language priors,” in Proc. ECCV. Cham, Switzerland: Springer, 2016, pp. 852–869.

[13] X. Liang, L. Lee, and E. P. Xing, “Deep variation-structured reinforcement learning for visual relationship and attribute detection,” in Proc. IEEE Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017.

[14] R. Hu, M. Rothbach, J. Andreas, T. Darrell, and K. Saenko, “Modeling relationships in referential expressions with compositional modular networks,” in Proc. IEEE Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 4418–4427.

[15] R. Yu, A. Li, V. I. Morariu, and L. S. Davis, “Visual relationship detection with internal and external linguistic knowledge distillation,” in Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (ICCV), Venice, Italy, Oct. 2017, pp. 1086–1096, doi: 10.1109/ICCV.2017.121.

[16] B. Zhuang, L. Liu, C. Shen, and I. Reid, “Towards context-aware interaction recognition for visual relationship detection,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017.

[17] Y. Zhan, J. Yu, T. Yu, and D. Tao, “On exploring undetermined relationships for visual relationship detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019.

[18] H. Zhang, Z. Kyaw, S. Chang, and T. Chua, “Visual translation embedding network for visual relationship detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Honolulu, HI, USA, Jul. 2017, pp. 3107–3115, doi: 10.1109/CVPR.2017.331.

[19] A. Mallya and S. Lazebnik, “Learning models for actions and person-object interactions with transfer to question answering,” CoRR, vol. abs/1604.04808, pp. 414–428, Oct. 2016. [Online]. Available: http://arxiv.org/abs/1604.04808

[20] Y.-W. Chao, Z. Wang, Y. He, J. Wang, and J. Deng, “HICO: A benchmark for recognizing human-object interactions in images,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015.

[21] X. Lin, Q. Zou, and X. Xu, “Action-guided attention mining and relation reasoning network for human-object interaction detection,” in Proc. 29th Int. Conf. Artif. Intell., Jul. 2020, pp. 1100–1104.

[22] S. Wang, K.-H. Yap, J. Yuan, and Y.-P. Tan, “Discovering human interactions with novel objects via zero-shot learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 11652–11661.

[23] T. Gupta, A. Schwang, and D. Hoiem, “No-frills human-object interaction detection: Factorization, layout encodings, and training techniques,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 9577–9585.

[24] Y.-L. Li, S. Zhou, X. Huang, L. Xu, Z. Ma, H.-S. Fang, Y. Wang, and C. Lu, “Transferable interactiveness knowledge for human-object interaction detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 3580–3589.

[25] J. Feyre, J. Sivic, I. Laptev, and C. Schmid, “Detecting unseen visual relations using analogies,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 9891–9900.

[26] B. Wan, D. Zhou, Y. Liu, R. Li, and X. He, “Pose-aware multi-level feature network for human object interaction detection,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 9469–9478.

[27] T. Wang, R. M. Anwer, M. H. Khan, F. S. Khan, Y. Pang, L. Shao, and J. A. W. Valente, “Deep contextual attention for human-object interaction detection,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 5694–5702.

[28] B. Xu, J. Li, Y. Wang, Q. Zhao, and M. S. Kankanhalli, “Interact as you intend: Intention-driven human-object interaction detection,” IEEE Trans. Multimedia, vol. 22, no. 6, pp. 1423–1432, Jun. 2019.

[29] B. Xu, Y. Wang, J. Li, Q. Zhao, and M. S. Kankanhalli, “Learning to detect human-object interactions with knowledge,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019.

[30] P. Zhou and M. Chi, “Relation parsing neural network for human-object interaction detection,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 843–851.

[31] C. Gao, Y. Zou, and J.-B. Huang, “ICAN: Instance-centric attention network for human-object interaction detection,” in Proc. BMVC, 2018.
[98] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov, “Learning word vectors for 157 languages,” in Proc. Int. Conf. Lang. Resour. Eval. (LREC), 2018. [Online]. Available: https://aclanthology.org/L18-1550/ 

[99] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “RoBERTa: A robustly optimized BERT pretraining approach,” 2019, arXiv:1907.11692. [Online]. Available: http://arxiv.org/abs/1907.11692 

[100] O. Levy, Y. Goldberg, and I. Dagan, “Improving distributional similarity with lessons learned from word embeddings,” Trans. Assoc. Comput. Linguistics, vol. 3, pp. 211–225, Dec. 2015. 

[101] T. Lan, W. Yang, Y. Wang, and G. Morì, “Image retrieval with structured object queries using latent ranking SVM,” in Proc. ECCV, 2012, pp. 129–142. 

[102] G.-J.-M. Kruijff, H. Zender, P. Jensfelt, and H. I. Christensen, “Situated dialogue and spatial organization: What, where . . . and why?” Int. J. Adv. Robotic Syst., vol. 4, no. 1, p. 16, Mar. 2007. 

[103] C. Galleguillos, A. Rabinovich, and S. Belongie, “Object categorization using co-occurrence, location and appearance,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2008, pp. 1–8. 

[104] W. Li, P. Zhang, L. Zhang, Q. Huang, X. He, S. Lyu, and J. Gao, “Object-driven text-to-image synthesis via adversarial training,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 12174–12182. 

[105] Y. Li, Y. Zhang, and Z. Zhu, “Error-tolerant deep learning for remote sensing image scene classification,” IEEE Trans. Cybern., vol. 51, no. 4, pp. 1756–1768, Apr. 2020. 

[106] G. Platonov and L. Schubert, “Computational models for spatial prepositions,” in Proc. 1st Int. Workshop Spatial Lang. Understand. New Orleans, LO, USA: ACL, Jul. 2018, pp. 21–30. 

**GUILLLEM COLLELL** received the B.S. degree in psychology and the B.A. degree in mathematics from the Autonomous University of Barcelona (UAB) and the M.S. degree in neuroeconomics from Maastricht University, in 2015. He is currently pursuing the Ph.D. degree with KU Leuven. His Ph.D. research focuses on learning joint representation for vision and language data, with an emphasis on spatial understanding. His work has been published in numerous top-tier conferences and journals, including ACL, AAAI, and TACL. He received the Best Paper Honorable Mention Award in ACL 2018 for his work investigating fundamental properties of neural network cross-modal mappings between visual and linguistic distributed representations.

**THIERRY DERUYTTERE** received the master’s degree in artificial intelligence from KU Leuven, in 2018, where he is currently pursuing the Ph.D. degree in artificial intelligence. He is working on creating an interactive agent that can execute commands, which can be used both in self-driving vehicles and medical robots. Some of his work has been published at EMNLP-IJCNLP2019 and Engineering Applications of Artificial Intelligence. He is the Main Organizer of the C4AV Workshop at ECCV 2020.

**MARIE-FRANCINE MOENS** received the M.Sc. and Ph.D. degrees in computer science from KU Leuven. She is currently a Full Professor with the Department of Computer Science, KU Leuven. She is the Director of the Language Intelligence and Information Retrieval (LIIR) Research Lab and the Head of the Informatics Section. Her main research interests include development of novel methods for automated content recognition in text and multimedia using statistical machine learning and exploiting insights from linguistic and cognitive theories. She holds the ERC Advanced Grant CALCULUS (2018–2023) granted by the European Research Council. From 2012 to 2016, she was the Coordinator of the MUSE Project financed by Future and Emerging Technologies (FET).