Intégration de Connaissances aux Modèles Neuronaux pour la Détection de Relations Visuelles Rares

François Plesse

UNIVERSITÉ PARIS-EST

Ecole Doctorale n°532 : MSTIC - Mathématiques et STIC

Thèse de doctorat

Spécialité Mathématiques

présentée et soutenue publiquement par

François PLESSE

le 27 février 2020

Intégration de Connaissances aux Modèles Neuronaux pour la Détection de Relations Visuelles Rares

Directeur de thèse : **Françoise PRÊTEUX**

Co-encadrement de la thèse :

Bertrand DELEZOIDE

Alexandru GINSCA

Jury

Patrick GALLINARI, Sorbonne Université	Président
Céline HUDELOT, CentraleSupélec	Rapporteur
Titus ZAHARIA, Télécom SudParis	Rapporteur
Gabriela CSURKA, Naver Labs Europe	Examinateur
Alexandru GINSCA, ATOS	Examinateur
Bertrand DELEZOIDE, CEA, LIST	Examinateur
Françoise PRÊTEUX, Ecole des Ponts ParisTech	Directeur

Je souhaite en tout premier lieu remercier Françoise Prêteux, Bertrand Delezoide et Alexandru Ginsca qui ont encadré ma thèse durant ces trois années. A Alexandru de m'avoir transmis sa passion pour le Machine Learning et m'avoir apporté son aide et ses idées pour la réalisation des expériences au jour le jour et la rigueur dans la rédaction, à Bertrand qui a guidé mes recherches, m'a partagé sa vision, et relu rigoureusement ce manuscrit. Et à Françoise qui a beaucoup travaillé pour me transmettre sa concision, rigueur et pédagogie dans la présentation de mon travail. Ce fut un véritable plaisir de travailler sur ces sujets avec des encadrants experts de ces domaines à mes côtés.

Je souhaite ensuite remercier Céline Hudelot et Titus Zaharia d'avoir accepté de rapporter sur ce manuscrit, et de m'avoir permis d'améliorer la qualité du manuscrit et de ma soutenance par leurs critiques pertinentes. Un grand merci également à Patrick Gallinari et Gabriela Csurka d'avoir fait partie de mon jury de thèse. Un grand merci également à mes compagnons de thèse Nhi, Othman, Youssef, Umang, Yannick, Eden, Jessica et Dorian, pour toutes nos discussions et nos partages aux cours de ces années communes au CEA. Plus généralement, merci au reste de l'équipe du LASTI, dont Hervé, Adrian, Olivier et Olivier, Romaric, Gaël, Benjamin. Merci à Odile, la secrétaire du LASTI pour son soutien qui m'a permis de m'en sortir pour toutes mes démarches.

Merci à Nikolas, mon colocataire qui m'a soutenu pendant les moments de doutes et pour fêter les moments de réussite. Un grand merci à tous mes autres amis qui m'ont motivé à me lancer dans cette voie : Thomas, Quentin, Cécile, Vivien, Pauline, Alexandre, Martin ainsi que ceux que j'ai malheureusement oublié de mentionner... Enfin, je souhaite remercier les véritables pilliers de cette thése : ma Maman, qui nous a quittés peu avant de pouvoir célébrer ce moment ensemble, mon Papa, mes frères et soeurs et Fabien mon compagnon. Merci pour votre curiosité à chaque fois que vous me demandiez ce que je faisais en ce moment, vos encouragements et votre soutien à toute épreuve pendant ces trois années. Cette thèse vous est dédiée et je vous souhaite bon courage pour sa lecture !

Intégration de Connaissances aux Modèles Neuronaux pour la Détection de Relations Visuelles Rares

François Plesse

Résumé

Les données échangées en ligne ont un impact majeur sur les vies de milliards de personnes et il est crucial de pouvoir les analyser automatiquement pour en mesurer et ajuster l'impact. L'analyse de ces données repose sur l'apprentissage de réseaux de neurones profonds, qui obtiennent des résultats à l'état de l'art dans de nombreux domaines. En particulier, nous nous concentrons sur la compréhension des intéractions entre les objets ou personnes vivibles dans des images de la vie quotidienne, nommées relations visuelles.

Pour cette tâche, des réseaux de neurones sont entraînés à minimiser une fonction d'erreur qui quantifie la différence entre les prédictions du modèle et la vérité terrain donnée par des annotateurs.

Nous montrons dans un premier temps, que pour la détection de relation visuelles, ces annotations ne couvrent pas l'ensemble des vraies relations et sont, de façon inhérente au problème, incomplètes. Elle ne sont par ailleurs pas suffisantes pour entraîner un modèle à reconnaître les relations visuelles peu habituelles.

Dans un deuxième temps, nous intégrons des connaissances sémantiques à ces réseaux pendant l'apprentissage. Ces connaissances permettent d'obtenir des annotations qui correspondent davantage aux relations visibles. En caractérisant la proximité sémantique entre relations, le modèle apprend ainsi à détecter une relation peu fréquente à partir d'exemples de relations plus largement annotées.

Enfin, après avoir montré que ces améliorations ne sont pas suffisantes si le modèle annote les relations sans en distinguer la pertinence, nous combinons des connaissances aux prédictions du réseau de façon à prioriser les relations les plus pertinentes.

Mots Clefs

Vision par Ordinateur, Interprétation Sémantique, Apprentissage Profond, Réseaux de Neurones Convolutifs, Détection de Relations Visuelles, Biais de Sélection, Connaissances Externes, Modélisation Sémantique, Pertinence

Knowledge Integration into Neural Networks for the purposes of Rare Visual Relation Detection

François Plesse

Short abstract

Data shared throughout the world has a major impact on the lives of billions of people. It is critical to be able to analyse this data automatically in order to measure and alter its impact. This analysis is tackled by training deep neural networks, which have reached competitive results in many domains. In this work, we focus on the understanding of daily life images, in particular on the interactions between objects and people that are visible in images, which we call visual relations.

To complete this task, neural networks are trained in a supervised manner. This involves minimizing an objective function that quantifies how detected relations differ from annotated ones. Performance of these models thus depends on how widely and accurately annotations cover the space of visual relations.

However, existing annotations are not sufficient to train neural networks to detect uncommon relations. Thus we integrate knowledge into neural networks during the training phase. To do this, we model semantic relationships between visual relations. This provides a fuzzy set of relations that more accurately represents visible relations. Using the semantic similarities between relations, the model is able to learn to detect uncommon relations from similar and more common ones. However, the improved training does not always translate to improved detections, because the objective function does not capture the whole relation detection process. Thus during the inference phase, we combine knowledge to model predictions in order to predict more relevant relations, aiming to imitate the behaviour of human observers.

Keywords

Computer Vision, Image Understanding, Deep Learning, Convolutional Neural Networks, Visual Relation Detection, Human Reporting Bias, External Knowledge, Semantic Modelling, Relevance

Intégration de Connaissances aux Modèles Neuronaux pour la Détection de Relations Visuelles Rares

Résumé substantiel en Français

Grâce à de récents progrès qui augmentent considérablement la puissance de calcul, la vitesse de transfert [3] et le stockage de données ainsi que la diminution du prix des processeurs graphiques (GPU) [1], la quantité de données disponible croît très rapidement. 400 heures de vidéos étaient envoyées à YouTube chaque minute en 2015 et 300 millions d'images sont envoyées à Facebook chaque jour [4]. Ces données ont un impact majeur sur la vie de milliards de personnes. Il est par conséquent crucial de les analyser pour être en mesure de comprendre les changements sociétaux qui en résultent, de recommander du contenu pertinent ou d'étudier des marchés potentiels.

Le domaine dédié à l'aggrégation, au traitement et à l'analyse de ces données est appelé pour ces raisons "Big Data". Nous nous intéressons dans cette étude à cette dernière, plus particulièrement à l'analyse d'images. Celle-ci repose, de façon croissante depuis 2012, sur l'utilisation de réseaux de neurones profonds. En effet, Krizhevsky *et al.* [72] proposèrent un réseau de neurones, appelé AlexNet, atteignant une erreur top-5 de 15.3% sur ImageNet [27], 10.8 points inférieure à l'état de l'art.

Cette avancée a suscité de très nombreuses recherches dans le domaine de l'apprentissage profond, pour la recommendation de contenus multimedia [21], la prise de décision, le marketing en ligne, la traduction automatique, l'extraction de contenu, etc. De nombreux champs de recherche en explorent l'utilisation pour la découverte d'interaction protéine-protéine [138], la génération de vidéos [20], le contrôle d'agents capables de jouer à des jeux de plateau, où les humains restaient jusqu'alors invaincus [129] ou encore de contrôler des voitures autonomes [11]...

Ces dernières avancées apportent par ailleurs une meilleure compréhension du contenu des images, avec la classification d'images [72], la détection d'objets [118, 120], la réponse aux questions sur des images [39] et la génération de légendes [157]. Toutefois, les tâches nécessitant des raisonnements haut niveau resistent aux modèles profonds [94]. Des applications telles que l'analyse de contenus provenant de réseaux sociaux ou la conduite autonomes pourraient bénéficier de telles capacités de raisonemment. En effet, la reconnaissance d'actions réalisées par des piétons pour prévoir leur comportement ultérieur nécessite davantage que la détection des objets, comme illustré sur la Figure 0.0.1. Ces actions peuvent être déterminées notamment en tenant compte de leurs positions relatives et l'évolution de celles-ci au cours du

FIGURE 0.0.1 – La prédiction des futures positions des piétons nécessite de les détecter, comprendre leurs directions, leurs intéractions avec ce qui les entoure.

temps, ainsi que du contexte de la scène.

Dans cette thèse nous nous concentrons sur la compréhension de ces intéractions entre objets ou personnes présents dans une images, et plus généralement, aux relations qui les lient.

Definition 0.0.1 *Une relation décrit la manière dont deux personnes ou objets sont connectées ; les effets d'une personne ou d'un objet sur un(e) autre.*

Les graphes sont des représentations naturelles de l'ensemble des relations d'une image, les relations étant les arêtes reliant les noeuds representant les objets visibles de l'image. Nous proposons donc d'extraire les relations sous la forme d'un graphe de scène, comme illustré Figure 0.0.2.

L'exécution de cette tâche requiert l'extraction automatique de concepts abstraits à partir d'informations visuelles. Les différences d'angle de vue et de contexte, les possibilités d'occlusion ainsi que les différentes appellations, avec différents niveaux d'information, d'un même objet visuel sont la source d'une grande diversité

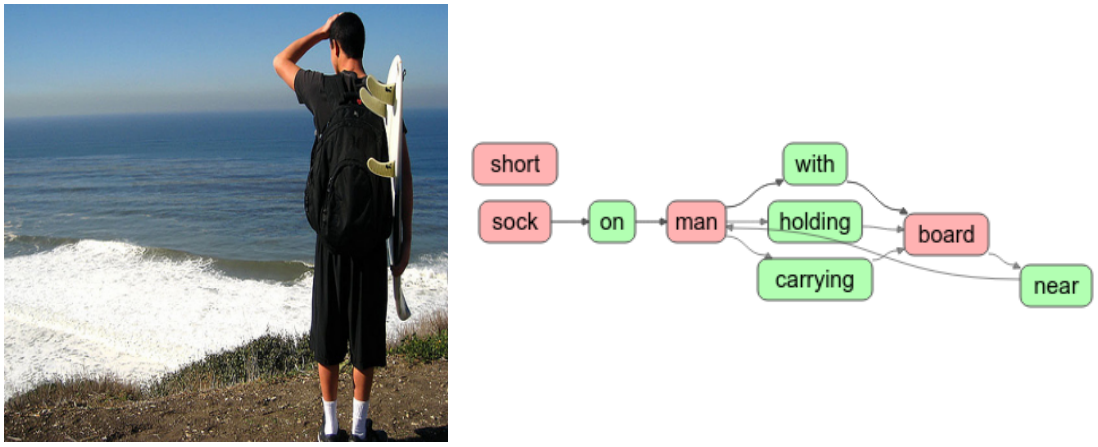


FIGURE 0.0.2 – Image extraite du Visual Genome [71] et le graphe de scène associé.

de représentations d'un même concept. Cela rend l'extraction de ces concepts difficile. L'extraction de relations est d'autant plus ardue que leur représentation dépend non seulement de celles des objets, mais aussi de leurs positions respectives. De plus, certaines relations sont polysémiques et synonynmiques entre elles, ce qui augmente la diversité de leurs représentations visuelles.

Dans cette thèse, nous nous intéressons particulièrement à l'extraction de relations par des réseaux de neurones profonds. Ces réseaux sont entraînés par apprentissage supervisé. Celui-ci consiste à optimiser une fonction qui caractérise la différence entre les prédictions du réseau et des annotations réalisées par des annotateurs humains. Les performances de modèles entraînés dépendent ainsi de la qualité de ces annotations, de leur nombre et de la part de l'espace des possibilités qu'elles couvrent et de la précision

Nous montrons que les annotations disponibles pour l'apprentissage de modèles d'extraction de relations ont tendance à être très déséquilibrées. Cela est la conséquence de plusieurs phénomènes. Tout d'abord, ce problème est de nature combinatoire, le nombre de relations possibles dans chaque image augmentant quadratiquement par rapport au nombre d'objets présents. Cela rend l'annotation exhaustive d'images très chronophage. Ainsi, les annotateurs doivent choisir des relations parmi l'ensemble des relations possibles. Ces choix ne sont pas complètement aléatoires, car ils dépedent des tailles, distances et positions des objets dans l'image, ainsi que des types d'objets concernés.

Motivés par ces observations, nous étudions l'impact de ce déséquilibre et montrons qu'il rend l'apprentissage de certaines classes difficile. Il limite par ailleurs l'évaluation des modèles entraînés, ne rendant pas compte de leur capacité à détecter l'ensemble des relations considérées. Pour y remédier, nous proposons de diminuer le besoin en exemples annotés en modélisant les relations sémantiques entre les classes de relations. Cette modélisation permet de caractériser la proximité sémantique entre relations et profiter des exemples d'une relation plus largement annotée pour apprendre à détecter une relation moins bien dotée. Enfin, la détections de paires d'objets à annoter est un aspect important de la génération du graphe de scène. Nous proposons d'entraîner un classifieur et de pondérer les scores de relations par le résultat de ce classifieur, résultat que nous appelons "pertinence" de la relation. Nous montrons que cela augmente la précision de la détection de relation, augmentant le nombre de vraies relations pour un faible nombre de prédictions.

Pour évaluer les modèles de détection de relations, nous les comparons sur plusieurs benchmarks : VRD [89], VG-IMP [158] et deux benchmarks que nous proposons, dérivés de Visual Genome [71]. Ceux-si sont notés VG-Large, avec plus de 10 000 classes de relations et VG-RMatters, que nous décrivons plus bas. Nous considérons deux tâches. La première, la classification de graphes de scène (SGCls),

consiste à classifier des régions d'une image et de détecter leurs relations. La seconde, la classification de relations (RelCls) consiste à détecter les relations à partir de régions déjà dotées de classes d'objets. La métrique utilisée est le rappel@k (R@k), où k est un nombre fixé de relations par image.

Biais de sélection de relations

Lors d'une première contribution, nous étudions le déséquilibre des classes de relation dans le benchmark le plus couramment utilisé, Visual Genome [71]. Nous montrons que ce déséquilibre (i) peut-être relié au processus d'annotation (ii) impacte l'apprentissage du réseau, entraînant un déséquilibre des relations prédites, très concentrées sur un faible nombre de classes (iii) empêche l'évaluation des modèles de rendre compte de leurs performances dans le cas général.

Nous proposons un réseau de référence, inspiré par plusieurs travaux [41, 166, 170] qui obtient des résultats compétitifs et permet d'évaluer l'impact de ce déséquilibre. L'extraction de relations est réalisée en plusieurs étapes. Un réseau de neurones convolutif extrait la représentation visuelle de régions correspondant à des détections d'objets et de paires d'objets. Les boîtes englobant les objets sont par ailleurs utilisées pour définir un masque binaire correspondant à la région de l'objet dans l'image, avec la valeur 1 à l'intérieur de la boîte et 0 à l'extérieur. Un nouveau réseau convolutif extrait ensuite la représentation de la configuration spatiale de chaque paire d'objet. Enfin, à partir des représentations visuelles des objets, de la paire d'objet et sa représentation spatiale, quatre branches sont entraînés à prédire des scores correspondant à chaque relation du vocabulaire.

L'apprentissage du réseau, c'est-à-dire la sélection des paramètres, est réalisé en optimisant la fonction ci-dessous par descente de gradient stochastique :

$$\begin{aligned} \theta &= \arg\max_{\theta} \mathcal{L}(\theta, \mathcal{D}) \\ &= \arg\max_{\theta} \mathcal{L}_o(\theta, \mathcal{D}) + \mathcal{L}_r(\theta, \mathcal{D}) \end{aligned} \tag{1}$$

où $\mathcal{L}_o$ et $\mathcal{L}_r$ sont les fonctions d'erreur pour la classification des objets et relations, définies par l'entropie croisée entre la sortie du réseau et les annotations d'apprentissage.

Enfin, la génération de graphe de scène est réalisée en sélectionnant les k relations avec le plus grand score, défini par le produit des probabilités de classes d'objets et de relations.

Evalué sur Visual Genome, nous mettons en évidence que le rappel des classes plus rares est très faible, et que cela n'est pas reflété dans le rappel global élevé (88.7%). Nous proposons d'évaluer les modèles sur une métrique supplémentaire :

le rappel macro par classe, où le rappel est calculé séparément pour chaque classe puis moyenné. Enfin, nous proposons une nouvelle partition du Visual Genome, VG-RMatters, qui augmente la diversité de classes. L'extraction de relations sur cette partition est plus difficile et permet de mieux évaluer les performances des modèles dédiés à cette tâche. La Table 0.0.1 compare la diversité des relations entre la partition de Visual Genome la plus courante, VG-IMP, et VG-RMatters. La diversité de relations est mesurée par la proportion d'exemples de la classe majoritaire ainsi que l'entropie moyenne des relations par paire de catégories d'objets.

Partition	Proportion de la majorité	Entropie moyenne
VG-IMP [158]	0.62	0.55
VG-RMatters	0.44	0.68

Table 0.0.1 – Proportion d'exemples de la relation majoritaire et entropie dans VG-IMP [158] and VG-RMatters pour les 50 paires d'objets les plus fréquentes. VG-RMatters a une plus grande diversité de relations.

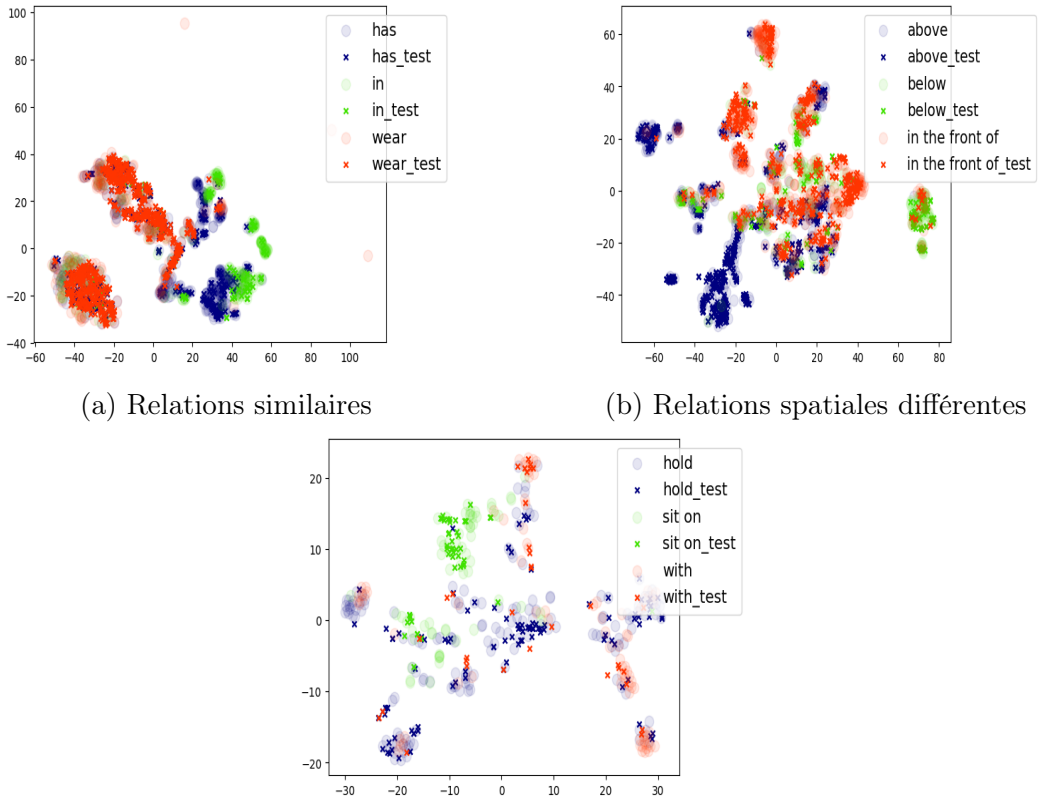
Modélisation sémantique pour l'apprentissage de relations rares

Dans une deuxième contribution, nous relaxons une hypothèse couramment faite dans l'apprentissage de modèles de détection de relations. Dans de nombreux travaux récents [58, 84, 154, 158, 164, 166, 170], les relations sont supposées mutuellement exclusives de façon implicite. Ainsi nous proposons plusieurs méthodes pour apprendre la représentation de relations en considérant leur relations entre elles.

Plus particulièrement, nous entraînons dans un premier temps un réseau dédié à l'extraction de représentations de relations dans un espace métrique, c'est-à-dire un espace doté d'une notion de distance. Contrairement aux travaux qui réalisent l'apprentissage en calculant l'entropie croisée entre deux relations, nous quantifions l'adéquation entre les prédictions du modèle et les données par la distance entre deux paires d'objets correspondant à une même relation. Ainsi nous pouvons relaxer les contraintes imposées au réseau et définir un espace dans lequel les paires d'objets correspondant à des relations similaires sont proches, comme illustré Figure 0.0.3

Cela nous permet par ailleurs de montrer les limites de la représentation apprise, car la différence entre deux relations peut-être due à une différence de configuration spatiales, d'objets, de contexte, etc...

Nous comparons cette méthode à des méthodes utilisant des données textuelles externes. Pour cela, un réseau de neurones est entraîné à respecter une contrainte sémantique. Cette contrainte est définie à partir de représentations de mots, et



(a) Relations similaires

(b) Relations spatiales différentes

(c) Actions différentes

FIGURE 0.0.3 – Représentation t-SNE [141] des relations dans l'espace métrique appris. Les croix correspondent à des relations de l'ensemble de test ; les cercles (mois opaques) représentent des relations stockées pendant l'apprentissage. Notre modèle est capable de regrouper des instances de la même relation, particulièrement les relations bien séparées, telles que "*above*" et "*below*".

quantifie la proximité entre deux classes de relations. Ainsi le modèle est entraîné à attribuer des probabilités similaires pour des classes apparaissant dans des contextes similaires. Nous montrons la pertinence de cette approche sur un dataset avec un très grand nombre de classes de relations, VG-LARGE et montrons qu'elle permet d'augmenter le rappel d'une approche de référence de 32%. La Table 0.0.2 résume ces résultats avec la comparaison de notre approche à deux approche à l'état de l'art sur deux benchmarks : VG-LARGE et VG-RMATTERS.

Pertinence de relations

Dans une dernière contribution, nous faisons l'observation que de nombreuses paires d'objets sont connectées par des relations non pertinentes, parce qu'elles sont trop courantes (par ex : un arbre a de l'écorce) ou peu intéressantes, car elles concernent des objets petits, distants, etc... Par ailleurs, la faible diversité des relations prédites par les modèles de détection de relation, due au déséquilibre de classes,

Benchmark	Approche	RelCls	
		R@100	R@100 Macro
VG-Large	IMP	32.7	-
VG-Large	Sémantique (Contribution)	**43.4**	-
VG-RMatters	MotifNet	87.8	46.6
VG-RMatters	Sémantique (Contribution)	**88.2**	**50.3**

Table 0.0.2 – Résultats avec notre approche de modélisation de relations sémantiques. Notre contribution présente le plus d'intérêt, augmentant signficativement le rappel pour un très grand nombre de classes, tel que VG-Large, et pour les classes rares, comme pour VG-RMatters.

nous motive à concentrer les prédictions sur les relations pertinentes de l'image, pour en augmenter la diversité.

La prédiction de la pertinence d'une relation est difficile, car elle dépend de nombreux facteurs connus (taille, distance, objets ...) et inconnus (liés au processus d'annotation de l'ensemble d'images, liés à l'annotateur ...). Nous proposons donc d'utiliser un classifieur de pertinence, correspondant à la probabilité qu'au moins une relation est annotée. Le score du classifieur est par suite moyenné à un potentiel basé sur des données statistiques mesurées sur l'ensemble d'apprentissage. Ce potentiel permet d'utiliser les relations entre variables dans les prédictions du modèle mais augmente les biais du modèle.

Cette contribution augmente de façon significative le rappel ainsi que le rappel macro de classe, comme le rapporte la Table 0.0.3

	SgCls	RelCls	
	R@20	R@20	R@20 macro
MotifNet [166]	38.0	58.4	19.8
Pertinence (Contribution)	**41.0**	**62.3**	**23.9**

Table 0.0.3 – Résultats avec pertinence de relation sur VG-RMatters. Le rappel est mesuré sur des graphes de scène avec 20 détections de relation.

VG-RMatters est un benchmark difficle, avec des relations variées pour chaque paire d'objets. Pour les images de celui-ci, notre modèle est capable de prédire une plus grande diversité de relations que les modèles à l'état de l'art et d'accroitre le nombre de vraies relations détectées.

Conclusion

La détection de relations visuelles est une étape importante pour comprendre et analyser automatiquement des images. Elle dépend fortement de la qualité de la répresentation des relations. Celles-ci dépendent de la qualité de représentations des objets, ainsi que des dépendances entre représentations visuelles, spatiales et sémantiques. Elles nécessitent donc un grand nombre d'exemples annotés pour les différencier dans des contextes similaires.

Cependant, la nature combinatoire du problème et le biais de sélection de relations rendent le nombre d'annotations déséquilibré en faveur d'un faible nombre de classes. Cela biaise les modèles appris sur ces données et ne permet pas d'évaluer les performances d'un modèle appliqué à de nouvelles images. Par ailleurs, la faible quantité d'annotation pour un grand nombre de relations ne permet pas d'apprendre à détecter ces relations.

Nous proposons une approche permettant d'intégrer des connaissances externes aux réseaux de neurones profonds, permettant de modéliser les relations entre classes de relations. Le modèle appris est ainsi capable de séparer les relations dans des régions prenant en compte ces similarités et d'utiliser un plus grand nombre d'exemples pour l'apprentissage de chaque classe.

Nous avons par ailleurs montré que la présence de la relation dans l'image n'est pas la seule information à considérer lors de la génération de graphes de scènes. Nous proposons d'intégrer la pertinence de cette relation au processus de génération des graphes de scènes. Ce processus se rapproche ainsi davantage au comportement humain. Par ailleurs, la diversité des relations prédites dans les graphes ainsi généres augmente, concentrant les prédictions sur les relations les plus pertinentes. Cela a pour effet d'augmenter le rappel des classes les plus rares, surpassant les méthodes à l'état de l'art sur plusieurs datasets.

Ces contributions obtiennent des résulats compétitifs, comme le montre la Table 0.0.4. La modélisation de relations sémantiques est la plus pertinente dans le cas d'un grand nombre de classes, où chaque classe est très similaire à un grand nombre d'autres classes, permettant d'apprendre chaque classe à partir d'un nombre d'exemples beaucoup plus important. Par ailleurs, l'intégration de la pertinence dans la génération de graphes de scènes est l'approche la plus impactante, soulignant le fait que les modèles de l'état de l'art ne capturent pas ou peu cette information.

Benchmark	Tâche	Etat de l'Art	Contribution	Résultats	Gain
Rappel					
VRD-set [89]	RelCls R@20	81.9 [165]*	Sémantique	80.8	-1%
VG-IMP [158]	SGCls R@20	37.6 [166]	Semantique	37.2	-1%
VG-IMP [158]	RelCls R@20	66.6 [166]	Pertinence	66.7	0%
VG-Large	RelCls R@50	22.7 [158]**	Semantique + Pertinence	45.2	99%
VG-RMatters	SGCls R@20	38.0 [166]**	Pertinence	41.0	8%
VG-RMatters	RelCls R@20	58.4 [166]**	Pertinence	62.3	7%
Rappel Macro					
VG-IMP [158]	RelCls R@100	37.9 [166]	Pertinence	44.4	17%
VG-RMatters	RelCls R@20	46.6 [166]**	Pertinence	52.6	13%

Table 0.0.4 – Principaux résultats de la thèse. * sont des réimplémentations et ** sont calculées avec des implémentations mise à disposition par les auteurs. La modélisation de relations sémantiques est la plus pertinente dans le cas d'un grand nombre de classes. L'intégration de la pertinence dans la génération de graphes de scènes est l'approche la plus impactante dans la majorité des cas. Cela souligne le fait que les modèles de l'état de l'art ne capturent pas ou peu cette information.

Knowledge Integration into Neural Networks for the purposes of Rare Visual Relation Detection

François Plesse

Abstract

Thanks to recent advances in computational power with a steep decrease in the price of Graphical Processing Units and increase in computations per second, data transfer speeds and data storage, more and more data is available. 400 hundred hours of video were uploaded to YouTube every minute in 2015 and 300 million images to Facebook every day. This data has a major impact on the day to day lives of billions of people, and it is critical to understand it in order to comprehend social changes, to recommend relevant content or recognize market opportunities. However, the amount of data makes it impossible for humans to manually extract information out of this content.

This thesis focuses on the automatic analysis of images. Recent advances in object detection have resulted in a sky-rocketting number of applications that rely on understanding the content of an image. However, a thorough comprehension of image content demands a complex grasp of the interactions that may occur in the natural world. The key issue is to describe the visual relations between visible objects. We tackle here the detection of such relations. We call this task **Visual Relation Detection (VRD)**. Many existing methods tackle this task by training deep neural networks with annotated images. This approach is hindered by the gap between visual and semantic representations, whereby visual and spatial representations of one relation have high variability. Indeed these representations depend on the angle of view of the image, the context, lighting and especially the objects involved in the relation. Additionally, synonymy and polysemy of relations increases this variability.

In light of these issues, we argue that visual information is not sufficient to learn to discriminate between relations. By considering the additional knowledge of relation similarities and focusing on relevant relations, they can be alleviated. Specifically, the major contributions of this work are as follow:

— **Human reporting bias in VRD datasets**: We show that VRD datasets have exploitable biases that are not apparent due to the used evaluation metrics. These biases come mostly from a high imbalance in available annotated examples, and a high dependency between objects involved in the relation and the corresponding relation class. We show how this impacts the detections of a competitive baseline and propose a metric as well as a new dataset

to better evaluate the performance of existing methods.

— **Overcoming Relation Imbalance with Semantic Modelling**: a VRD model is trained so that relations that are similar have similar probabilities. Two methods are proposed: the first method relies on a k Nearest Neighbor approach to train a deep neural network, in order to improve uncommon relation classification and take into account the structure of relations. For the second, the standard supervision is augmented with additional constraints from text data, in order to reduce the model bias and increase model generalization.

— **Relations Relevance**: A new scene graph construction method is introduced, integrating a learnt relevance criterion. In the absence of annotations for this criterion, two methods are proposed in order to focus on object pairs frequently related in similar contexts. The first relies on self-supervision and the second on high-level dependencies between concepts. The impact of these methods is analyzed showing that the constructed scene graphs contain more uncommon relations while keeping a high overall recall and thus reduces the impact of the reporting bias. Furthermore, we find that this additional factor allows our model to predict relations on fewer and more relevant object pairs.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Thanks to recent advances in computational power, with a steep decrease in the price of Graphical Processing Units (GPUs) [1], in data transfer speeds [3] and data storage, more and more data is available. For instance, four hundred hours of video were uploaded to YouTube every minute in 2015 and 300 million images to Facebook every day [4]. This data has a major impact on the day to day lives of billions of people, and it is critical to understand it in order to comprehend social changes, to recommend relevant content [21] or recognize market opportunities.

Thus deep learning models have been adopted in many different sectors, such as recommendations for multimedia services or electronic stores. They are also increasingly used for decision making, online marketing, automatic translation, content extraction. Many research fields explore their use for protein-protein interaction [138], video generation [20], agents able to play games [129], to control self-driving cars [11] and so on.

However, these models present several noticeable flaws. Indeed, deep learning models are much less efficient at generalization and learning complex rules than humans, as shown in [76, 77], requiring hundreds of labeled examples to learn new concepts. In [94], Marcus shows that models trained by reinforcement learning have a very shallow understanding of the space with which the agent interacts. After being trained for several hundred hours on a single game, their performance can be thoroughly undermined with small perturbations. Furthermore, when it comes to text understanding, recurrent neural networks do not adapt well to differences in test and training test that require compositional skills. This is in part due to the representation of sentences as word sequences whereas language is intrinsically hierarchically structured, as argued by linguist Noam Chomsky [19].

These limitations are symptoms of the fact that deep learning models are very good at finding correlations between variables, whether these variables are features of images, text documents or sound... However, these correlations are not always always related causally, *i.e.* one variable is not the cause of the other and vice versa. For example, Ribeiro *et al.* [121] show evidence that, when trained on a small dataset with pictures of wolves and huskies, the wolf classifier only learns to detect snow. Features associated with snow textures are the most statistically distinguishing feature but this relationship is not causal.

Furthermore, recent studies [31, 45] suggest that some non-causal relationships learnt by deep learning models arise from a phenomenon whereby models latch on high-frequency information. They show that a model is able to recognize images

passed through a high-pass filter, recognizing images that are nearly invisible to humans. This makes it sensitive to additive high-frequency noise, which explains why adversarial examples are so efficient at fooling deep neural networks.

The latter limitations can be tackled by adversarial training, while the former, *i.e.* learning spurious correlations between low-frequency features, mainly have implications in specific settings. Such settings tend to have small datasets or strongly differing training and testing distributions (*e.g.* if all wolves in train are in snowy contexts but not in test). With the advent of large datasets such as ImageNet [27] and transfer learning techniques [106], the drawbacks of small datasets can be mitigated. Many solutions have been proposed accordingly, on the one hand by integrating external knowledge, in the form of class attributes for zero-shot learning [78, 108, 155] and on the other hand by bringing together symbolic and neural modelling. The latter works involve discrete operations with neural models [101], use a combination of reinforcement learning that learns to translate sentences into symbolic programs and supervised learning to learn scene representations [92] for Visual Question Answering.

Thus, despite large strides in image understanding with tasks such as image classification and object detection, neural models perform poorly at tasks that do not only require recognition but also higher level reasoning [94]. Applications such as autonomous driving and social media analysis for purposes of epidemiology or market analysis could benefit such reasoning. To recognize the action in which pedestrians are engaged in, or how many people are smoking in a picture, one needs not only detect all visible objects but also how they interact. These interactions depend on the relative positions of objects and people and their respective parts, as well as the context around them.

1.2 Basic Concepts and Issues related to Visual Relation Detection

Let us introduce several basic concepts on which this work is based.

Definition 1.2.1 *A concept is a general and abstract idea of a concrete or abstract object made by a human mind, allowing it to connect to it perceptions and organize knowledge.*

This definition adapted from Larousse Dictionary underlines the important connection between perception and knowledge that are necessary to understand the

world. Concepts can be partitioned in several categories: *objects*, *actions*, *scenes* and so on, as shown in Figure 1.2.1.

Figure 1.2.1 – Several types of concepts may be extracted from one image. Here: **objects**: *man, frisbee*, **actions**: *throwing frisbee*, **scene**: *outdoor sports.*

The detection of a type of high-level concepts which describes interactions, relative spatial configurations and possession, which we call relations, is the object of this dissertation.

Definition 1.2.2 *A relation describes the way in which two people or things are connected; the effect of a person/thing on another.*

Relations are the description of how objects interact, thus graphs are natural representations of these high-level concepts. Hence, we propose to extract such relations from still images in the form of scene graphs. Scene graphs are graphs comprised of object nodes, representing objects visible in the image and relationship edges, representing true relationships between the objects of the image. Two examples of such graphs are represented in Figure 1.2.2.

Tackling this task requires the automatic extraction of concepts from visual information. As mentioned in [6], this raises the problem of the semantic gap, which characterizes the differences between a concept and its representations in different modalities. Visual representations of a concept have high variability due to multiple factors: membership to different sub-concepts, angle of view, occlusion, context, lighting and so on. This is especially true when it comes to the relations between objects because their representations vary depending on perspective and especially depend on the involved objects.

Figure 1.2.2 – Two images from Visual Genome [71] and their respective annotated scene graphs.

Figures 1.2.3 and 1.2.4 show how this gap manifests for spatial relations (*e.g. on the left of*) (resp. actions (*e.g. fish in*)) with different viewpoints (resp. different object pairs). In these examples the underlying relation concept remains the same while both low level visual representation and spatial configurations are very different.

This semantic gap manifests in a second way as shown in Figure 1.2.5 due to the polysemy of relations.

Definition 1.2.3 *Polysemy is the phenomenon whereby a single word form is associated with two or several related senses.*

Polysemy of relations increases this gap by increasing the variability of the visual representation of relations. For example, with relation "*in*", the shape of the object one is "*in*" will imply very different spatial configurations as illustrated in Figure 1.2.5.

(a) parking meter on the left of skier (b) boy in red on the left of boy in black

Figure 1.2.3 – Relation "*on the left of*" for different viewpoints, *i.e.* where the photographer is behind the object pair (a) or in front of the object pair (b). The spatial configuration of the relation depends on the of position of the photographer relative to the objects.

(a) person fishing in sea (b) bear fishing in river

Figure 1.2.4 – Relation "*fish in*" for different object pairs. The visual representation of one relation is highly influenced by that of the involved objects.

(a) *dog in car* (b) *dog in shirt* (c) *woman in hat* (d) *woman wearing hat*

Figure 1.2.5 – On the one hand, a polysemous relation can have several meanings and thus have a high intra-variability, as in (a), (b) and (c). On the other hand, in several contexts, different relations can have the same meaning and have similar visual representations, as in (c) and (d): they are synonymous.

Definition 1.2.4 *Synonymy is the phenomenon whereby two different words are associated with the same sense.*

Synonymy does not participate in the semantic gap but increases the confusion between relations and directly ties into the problem of polysemy as relations can be synonymous in some contexts and have different meanings in others. For example

the relation "*in*" in *dog in shirt* is synonymous to "*wear*". However with the pair *(dog, car)*, this synonymy does not hold anymore.

1.3 Problem statement

Having defined these concepts, we formally define the task of Visual Relation Detection. We consider the problem of scene graph labelling, *i.e.* of defining a function that takes as input a real valued image and outputs a scene graph with labeled nodes and edges. Let f be a graph labelling function

$$f : \mathbb{R}^{h_I \times w_I} \to \mathbf{Seq}(\{0,1\}^{n_\mathcal{C}}) \times \mathbf{Seq}(\{0,1\}^{n_\mathcal{R}}) \tag{1.1}$$

where $\mathbf{Seq}(X)$ denotes the spaces of finite sequences in X and h_I and w_I are the dimensions of input images. The number of object and relations classes are referred to as $n_\mathcal{C}$ and $n_\mathcal{R}$, respectively.

Given an image I,

$$f : I \mapsto (\{\mathbf{v}_1, \dots, \mathbf{v}_n\}, \{\mathbf{e}_{1,1}, \dots, \mathbf{e}_{n,n-1}\}) \tag{1.2}$$

is a labeling of I and for all $i \in [1 \dots n]$ and $k \in [1 \dots n_\mathcal{C}]$

$$v_{i,k} = \begin{cases} 1 \text{ if } c_k \text{ is a label for object } i \\ 0 \text{ otherwise.} \end{cases} \tag{1.3}$$

and for all $h, t \in [1 \dots n], h \neq t, k \in [1 \dots n_\mathcal{R}]$

$$e_{h \to t,k} = \begin{cases} 1 \text{ if } r_k \text{ is a label for edge } (h,t) \\ 0 \text{ otherwise.} \end{cases} \tag{1.4}$$

This allows the mapping to assign multiple labels to one node or edge leading to a multi-label classification. We make this choice because several relations may be true for one pair of objects, especially spatial relations and actions, or different compatible actions as in Figure 1.3.1.

Furthermore, for each object pair (h, t), one relation may be true without being annotated by a human. Thus we propose to distinguish two tasks, with different challenges, which we explore in depth in this work:

1. **Relevance Classification** is the task consisting in predicting whether at least one relation is annotated, *i.e.* there exists $k \in [1 \dots n_\mathcal{R}]$ such that $e_{h \to t,k} = 1$

2. **Relation Classification** consists in predicting what relation is true, given that at least one is annotated

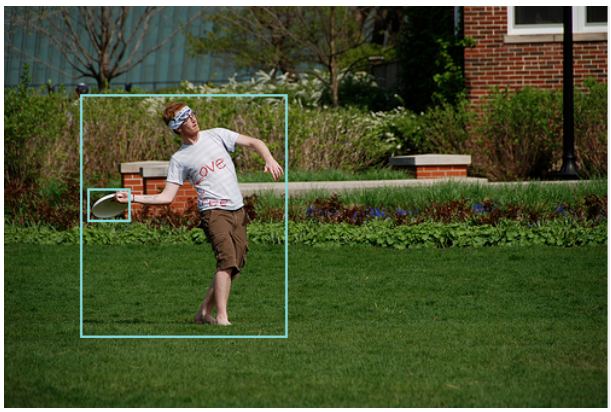


Figure 1.3.1 – Image example from Visual Genome [71]. Several relations are true between the selected objects: *(person, right of, frisbee), (person, holding, frisbee), (person, throwing, frisbee), (person, with, frisbee)...*

1.4 Report Outline and Contributions

The obstacles presented in Section 1.2 and Chapter 3 are the main motivation for the contributions of this work, summarized as follows.

— In a **first Chapter**, we study how several limitations of Deep Neural Networks impact these networks and how they pertain to the issues related to VRD. Then we show how they can be tackled and how they can be applied in the context of VRD.

— In a **second Chapter**, we show some limitations to existing VRD methods. We show that, due to the human reporting bias, the evaluation of the loss function does not accurately represent the true relation distribution and thus models do not generalize to new images. Furthermore, we show that the loss function does not correlate with performance on the target metric. Finally, having shown that evaluation of existing models on imbalanced datasets does not completely capture their performance, we propose a new metric and a new split of the most studied VRD dataset, in order to highlight these limitations. The definition of this split has been submitted to WACV 2020.

— The foremost limitation is the poor performance of the detection of rarer relations, which we tackle in a **third Chapter** by modelling semantic relationships between relation classes. We integrate semantic knowledge into the network during the training phase in order to improve the accuracy of the relation distribution estimation. We propose two methods for modelling semantic relations, which respectively bring a 13% and 33% relative increase

in recall averaged over classes (resp. images). These results have been presented in two separate publications [113, 114] and in an extended publication submitted to MTAP 2019.

— Finally, in a **fourth Chapter**, we show that the scene graph generation process is critical to the performance of our VRD model. Knowledge is here combined to the model predictions during inference, in order to tackle problems that are not captured by the training objective. Introducing a new relevance classifier, we can significantly increase the recall of relations, especially that of rarer relations, increasing the recall averaged over all classes by 13%. This contribution was introduced in our publication [113] and further proof of its impact is shown in our work submitted to WACV 2020.

Chapter 2

State of the Art

This chapter consists of a survey of the Visual Relation Detection (VRD) field. However, this is a very young field, therefore many existing methods that have been proposed in related fields such as object detection, image classification, few-shot learning, have not been applied to VRD. As all VRD competitive methods make use of Deep Neural Networks. Before diving into this task, we give a broader introduction of the training process of Deep Neural Networks and present some of these methods.

2.1 Neural Networks

Our goal is to extract from a still image a scene graph. To achieve this, we aim to define an image labeling function that takes as input any image and outputs this scene graph. This function can then be evaluated by several measures, such as the fraction of true relations in the output graph, the fractions of true relations in the image that are present in the graph and so on.

This task is part of a greater set of problems which can be described as finding a mapping $f : X \to Y$ between an input vector space X of dimension d_{in} and an output space Y. This mapping should satisfy a set of constraints. For example, in the case of VRD, the input space X is the set of possible images, Y the set of possible scene graphs.

Constraints on f would be that for each I in X and (o_1, r, o_2) in $f(I)$, objects o_1 and o_2 are present in I and the relationship triplet (o_1, r, o_2) is true. In this work, we consider relations between detected objects. Thus a relation between two objects is true if and only if

1. both objects are correctly classified
2. the relation between the objects holds true

2.1.1 Definition

When considering tasks such as image or text understanding, input and output spaces have several thousand dimensions. For images, this is due to the number of pixels and for texts, to the number of possible word combinations. In recent years, both the quantity of available data and computational power has increased very rapidly, making supervised learning approaches competitive in the task of deriving such mappings. These methods are based on training parametric models with annotated data so that the models are able to reliably predict labels for unanotated data.

Supervised Learning Let $\mathcal{D}$, a dataset comprised of input pairs $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ where for each $i \in [1 \ldots n]$, $(x_i, y_i) \sim p$ is drawn independently and identically distributed with a distribution $p : (X, Y) \to \mathbb{R}_+$.

Supervised learning is a subset of Machine Learning algorithms. The aim of these algorithms is to find a mapping over a specified space of mappings $\mathcal{F}$ that best fits the training data. The space $\mathcal{F}$ is defined as $\{f_\theta | \ \theta \in \mathbb{R}^d\}$: a set of parametric functions, parameterized by a vector of parameters θ.

For this, an objective function ϕ is defined to quantify the fit between a given sample $(x_i, y_i) \in \mathcal{D}$ and the output $\tilde{y_i} = f^*(x_i)$. Thus f^* is defined as

$$f^* = \arg\min_{f \in \mathcal{F}} \mathbb{E}_{(x,y)\sim p}[\phi(f(x), y)] \tag{2.1}$$

The search of this mapping is called training. As mentioned in [53], this makes training models different from optimization problems, where the performance measure is directly optimized. In contrast, neural networks are evaluated on another metric M. However, its optimization is usually not tractable, which is why a different object function is optimized, in the hope that it improves M.

In Equation 2.1, the mapping is defined as the function that minimizes the **risk**. In practice, this expectation is usually not available so the experimental risk is optimized instead:

$$f^* = \arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{(x,y)\in\mathcal{D}} \phi(f(x), y) \tag{2.2}$$

While the standard error of this estimation is equal to $\frac{\sigma}{\sqrt{n}}$ where σ is the standard deviation of $\{\phi(f(x), y)\}_{(x,y)\in\mathcal{D}}$ and decreases with the number of examples n, the computation of the gradient for equation 2.2 is linear with respect to n. in $\mathcal{D}$. For very large datasets, this becomes intractable and thus motivates the use of Stochastic Gradient Descent [67, 122], where the gradient of the function is estimated using a sample of the training set, called a **minibatch**.

Finally, the training process differs from the optimization process because of **early stopping**. Since the objective function is different from the target metric, an additional criterion is used to select a stopping point for the optimization process, typically evaluating the target metric M on a validation set defined beforehand.

As most competitive methods that tackle computer vision tasks, VRD included, are based on Neural Networks, we study in this section this class of mappings. They do not require to manually define a representation of the input data but instead directly use its raw representation. can be defined as the composition of a number of parametric functions such as affine functions and activation functions as illustrated on Figure 2.1.1. From this, we can define:

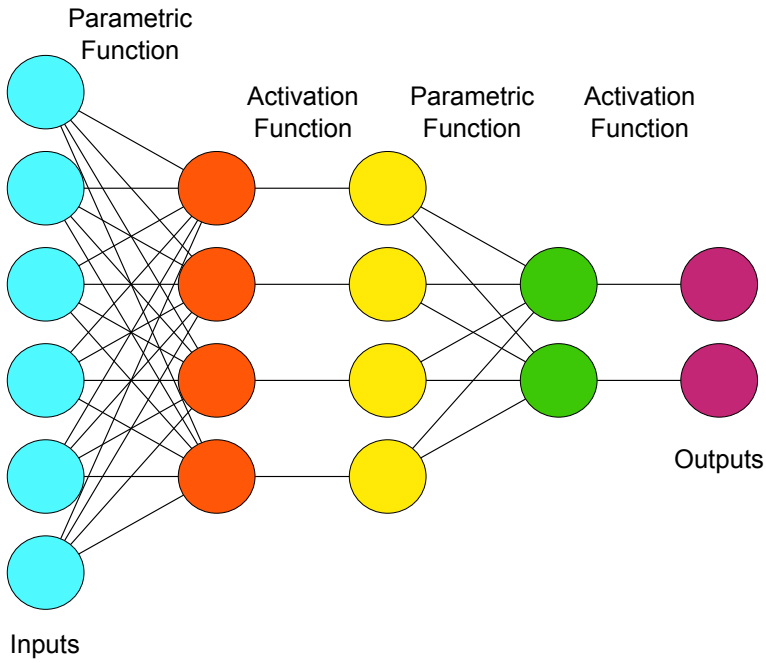


Figure 2.1.1 – Schema of Neural Network with two layers, taking a 6-dimensional input and outputting a two-dimensional vector. Each layer is defined as the composition of a parametric and activation function. Here both parametric functions are affine functions summing a weighted sum of inputs and a bias. They are followed by one activation function.

A Neural Network is a parametric mapping f defined as the composition of parametric functions and activation functions optimized to minimize an experimental risk on a set of inputs.

2.1.2 Classifiers

We specifically focus on the task of classifing inputs (*e.g.* images) into a set of classes $Y = \{1, \ldots, n_c\}$. Thus we consider classifiers, *i.e.* a class of mappings from a vector space X to a discrete finite space. To use Deep Neural Networks as classifiers, they are usually defined such that an intermediate representation of the input, noted x', is extracted. Then the last layer of the neural network is set as the affine function:

$$x' \to \tilde{y} = W \cdot x' + b \tag{2.3}$$

where $W \in \mathbb{R}^{d' \times n_c}$ and d' is the dimension of x'. The conditional probability of class i is then computed with the softmax function

$$q(y = i|\theta, x) = \frac{\exp \tilde{y}_i}{\sum_{k=1}^{n_c} \exp \tilde{y}_k} \tag{2.4}$$

where θ is the set of parameters. The model parameters are then defined as the parameters maximizing the log likelihood function:

$$\theta^* = \arg\max \sum_{(x,y)\in\mathcal{D}} \log q(y|\theta, x) \tag{2.5}$$

$$= \arg\max \sum_{(x,y)\in\mathcal{D}} \sum_{j=1}^{n_c} p(y=j|x) \log q(y=j|\theta, x) \tag{2.6}$$

$$= \arg\max -H(p,q) \tag{2.7}$$

$$= \arg\min H(p,q) \tag{2.8}$$

where H is the cross-entropy function.

2.1.3 Convolutional Neural Networks

Having defined how the output distribution is extracted, we now study how the intermediate representation is extracted. In this work, where the input is an image, this representation is extracted by convolutional layers. Indeed, Convolutional Neural Networks have reached competitive results when processing images, due to several properties which we explain after describing how these layers process inputs.

Convolutional Neural Networks (CNNs) are a class of neural networks where the parametric function are kernels which are convoluted over their input. Let f a convolution layer.

$$f(I)(i,j) = \sum_m \sum_n I(i-m, i-n)K(m,n) \tag{2.9}$$

where K is a two-dimensional kernel.

As mentioned in [53], the motivation behind these convolutional layers is **interaction sparsity**, **parameter sharing** and representation **equivariance to translation**. The first refers to the smaller size of the kernel when compared to the input, decreasing the number of parameters and the number of operations. The second refers to the sharing of parameters between different image positions. Contrary to feed-forward layers, the output of a single channel is parameterized by a single kernel for all image locations. Finally, equivariance to translation means that if the input is translated by a given amount, then the output is translated by the same amount.

Thus these networks are efficient with respect to the number of used parameters and are well adapted to large grid-structured inputs which is why they are very widely-used for image processing tasks.

2.1.4 Object detection

Object Detection is the first step towards VRD, as it is required to define the nodes in the image scene graph. An object detector is a mapping from an image input space X into the set of finite object sequences $Y = \mathbf{Seq}(R^4 \times \mathcal{C})$, where for each $y = (b, c) \in Y$, b corresponds to the coordinates of the object bounding box and c is in $\mathcal{C} = \{0, 1, \ldots, n_c\}$, a finite set of object classes.

Finding a mapping that minimizes a given objective function is a task that can not be solved by the same architectures as image classifiers as the number of outputs changes depending on the input. A naive approach would rely on first extracting a set of regions of interest then processing each part of the image bounded by a box and classify the presence of an object. However, this process is inefficient in computation time as well as memory as the number of necessary proposals to obtain a high detection score would be too high.

Regions with CNN features (R-CNN) To bypass this problem, Girshick et al. [48] proposed R-CNN, a model taking as input a set of 2,000 region proposals extracted with Selective Search [140], which extracts a representation of each region by passing them through a pre-trained CNN, and taking the output of a dense layer, then classifying each one using binary SVMs. Furthermore, a linear regression is trained to predict more accurate bounding box coordinates. The main drawback of this method are the processing of a high number of regions of interest by a neural network, resulting in a 47s average processing time per image.

Fast R-CNN This drawback is addressed by Girshick et al. [47] with the Fast R-CNN model. It is able to extract a representation of each region proposal by processing the image once using ROI-pooling with the same region proposals as in [48]. ROI-pooling consists of extracting a fixed-sized grid representation of each region proposal, by pooling the maximum value at each grid location. Since the ROI-Pooling operation is differentiable, the network can be trained from end-to-end, after a pre-training, which allows Fast R-CNN to outperform R-CNN from 62.4 to 65.7 mean Average Precision (mAP) on VOC 2012 [32] while decreasing the processing time from 47s to 2.3s per image at test time.

Faster R-CNN The last method has two drawbacks: the region proposal method is still separated from the object detection model, which makes errors at the proposal level impossible to correct. Furthermore, most of the processing time comes from the high number of region proposals of Fast R-CNN (2,000). Instead of using Selective Search, Ren et al. [120] define a sub-network called RPN (Region Proposal Network). A grid is defined over the extracted feature map of the image with a set of 9 anchors

at each point of the grid. After removing duplicate detections with Non Maximum Suppression (NMS), the number of region proposals is decreased to 300. They then define an error function to evalutate the quality of each object region detection used to update the model parameters. For each selected anchor, an objectness score and bounding box coordinates are output by a softmax classifier and the same regressor as in Fast R-CNN. The object detection is then used witht the same network as in Fast R-CNN, sharing convolutional layers with the RPN. With their proposed training procedure, the mAP increases from 65.7% to 67.0% without additional training data, and 75.9% with pre-training on MS COCO and Pascal VOC 07 [32], with a decreased processing time of 0.2s per image [120].

You Only Look Once You Only Look Once (YOLO) [119] changes the paradigm of object detection by removing the need of region proposals. Instead, the image is divided into a grid of fixed dimensions and the network predicts two bounding boxes at each grid location with a confidence score and class probabilities at this point, independently from the bounding box and in only one forward pass. Thus the whole image impacts the detection at each point and any predicted bounding box will have the most probable class of the corresponding grid point. This also makes the object detection much faster. Their final version [118] decreases the processing time to 0.025s per image with a mAP of 78.6%, higher than Faster R-CNN.

Method	AP
Faster R-CNN [120]	34.9
Yolov3 [118]	33.0

Table 2.1.1 – Object Detection Results on MS-COCO [86]

Table 2.1.1 shows results for object detection on MS-COCO [86]. Faster R-CNN is slower than YOLO but outperforms it on the task of object detection. We focus on Faster R-CNN because they have been shown to provide representations of images that can be used in diverse tasks (see Section 2.3.1, at the cost of processing speed.

2.1.5 Caption generation

The task of caption generation, which consists of finding a mapping from an image space to the space of finite sequences of words has garnered interest with applications from image retrieval [160] to automatic image description for visually impaired people [90]. It is similar to Visual Relation Detection, as both tasks consist in extracting higher order descriptions than detecting objects. The outputs of Caption Generation models however are unstructured, contrary to those of VRD models.

This makes captions more flexible but also less readily usable for downstream tasks, such as image retrieval.

Competitive models are comprised of two successive neural networks. One is a CNN acting as image encoder, representing the image in a much lower dimensional space. The second is a recurrent neural network decoding the encoded image into a set of sentences. Vinyals *et al.* [145] propose to use a CNN pre-trained as an image classifier on ImageNet [49] and an LSTM (Long Short-Term Memory) [60] network taking as context the image encoding and the previously predicted words. The network is then trained to maximize the log likelihood of the correct word. This method is extended in [159] with an attention mechanism (see Section 2.4.5) that attributes higher weights to parts of the image for the prediction of each word.

2.2 Metric learning

2.2.1 Motivation

The purpose of maximum likelihood classifiers as described in Section 2.1.2 is to find a set of hyperplanes that for each class separates instances of that class from instance of other classes. Recently, a change of paradigm has been proposed in order to enable the scaling of models to a high number of classes, learn from few examples and use distance-based methods. These methods aim to define a distance on the input space so that the distance between similarly labeled instances class is smaller than between instances with different labels.

2.2.2 Mahalanobis distance learning

The Mahalanobis distance between two random vectors of the same distribution with covariance matrix S is defined as $d_S(x, y) = \sqrt{(x-y)^T S^{-1}(x-y)}$. When decomposing the matrix S as $S = L^T L$, the Mahalanobis distance can be reinterpreted as the Euclidean distance between two linearly transformed vectors: $d_S(x, y) = \sqrt{(x-y)^T L^T L(x-y)} = \sqrt{(Lx-Ly)^T(Lx-Ly)}$.

Xing et al. [156] made the observation that clustering algorithms are reliant on having a good metric to find a meaningful set of clusters. They propose to learn this metric by using a family of Malanahobis distances parametrized by a matrix and minimize the distance between similarly labeled instances with regard to the matrix parameter. This method was expended upon by relaxing the constraint on the distances and removing the symmetric pseudo-definite constraint [127], by optimizing the leave-one-out error of a stochastic nearest neighbor classifier [51] or minimizing the distance between neighbours while keeping distance between non-neighbours greater than a margin [153]. However these optimization problems become intractable when dealing with a high-number of constraints which makes them inefficient for problems with many data points.

2.2.3 Triplet loss

When the number of samples becomes too high, the metric space described above becomes hard to learn, due to the high number of constraints. To remedy this, Weinberger et al. [153] introduce a triplet loss, which enforces instances of the same class to have a smaller distance than instances of different classes, by a margin. Contrary to Mahalanobis Metric Learning, this model does not require that the input distribution be normal or unimodal. Thus the network parameters are set

as

$$\theta^* = \arg\min_{\theta} \sum_{(x_+,y_+)\in\mathcal{D}} \sum_{(x_-,y_-)\in\mathcal{D}|y_-\neq y_+} \max(\alpha - \|x_+ - x_-\|, 0) \tag{2.10}$$

They show improved performance on several tasks such as face recognition when compared to SVMs but also other kNN classifiers with common dimensionality reduction methods such as PCA [110] and LDA [37].

2.2.4 Deep Metric Learning

This change has been adapted to Deep Neural Networks to learn from few examples [132, 146] and even zero examples of a given class [65], aiming to learn representations that group together elements of the same classes.

Schroff *et al.* [126] train a network to output similar representations of the face of the same person in different settings. They extract the L2-normalized representation of a person's face. At training time, negative samples that are more distant than positive example are used in order to avoid a collapsed model (when all points are aggregated near the origin). For fine-grained recognition, Cui *et al.* [22] select anchors by K-Means after metric learning and have them vote on the class of new images based on their distance to the image. Furthermore, the classification loss from these anchors is incorporated during the training phase. Sohn [134] makes the observation that deep neural networks trained with a triplet margin loss have a slow convergence and get unstable towards the end of training. They attribute these problems to the comparison of each sample to a single negative example, which have a high probability of yielding a zero loss. They propose to instead select a fixed set of N negative samples, each of a different class, for each set of batch samples. Each batch sample has only one positive sample and the positive pair is then compared to the $N-1$ negative pairs instead of one. Similarly, in [135], Song *et al.* train a model to embed images from different perspectives with a high similarity by comparing each pair of images of the same object to several hard negatives.

Finally, Kaiser *et al.* [65] define a memory containing keys and values for each class. This set is updated during the training phase to represent the whole set of training samples. The closest negative and positive keys are used to compute the model loss. This keeps the set of examples small and defines a set of class representatives, even with few examples per class, that new samples are compared to.

At the intersection between Metric Learning and Hierarchical Modelling (Section 2.3.4), Nickel and Kiela [105] enforce constraints between WordNet [34] concepts by embedding them in a hyperbolic space. They show that similarities between concepts are learned directly from hierarchies and that it is possible to predict links between

Method	Description
FaceNet [126]	Face recognition with soft hard negatives
Anchor classification [22]	K-means on image embeddings and classifiation from centroids
N-pair [134]	One negative example per class
Lifted Structured Embedding [135]	Several hard negatives per pair of objects from the same class
Rare Events [65]	Prototypes represent classes from which Positive and Hard Negative are drawn
Poincaré Embeddings [105]	Embed concepts in a hyperbolic space

Table 2.2.1 – Deep metric learning

unseen concepts reliably, requiring a lower space dimension than euclidean distances.

Table 2.2.1 summarizes the several strategies for learning a metric space. The main area of focus is the example selection, especially since pairs of samples are compared to each other and the selection of the pairs will have a strong impact on the resulting space.

2.2.5 Metric Learning for few-shot learning

Due to some specificities of the available data, shown in Chapter 3, we study specifically the performance of networks trained to learn with few examples. For this, we compare several methods that classify images from how similar they are to annotated images. For this, they compare raw pixels, train a model to discriminate whether two examples come from the same class [70], to predict attention to examples from a support set [147] and to extract memory keys representing annotated classes [65]. Table 2.2.2 shows the comparison of these methods on Omniglot [75], a dataset with hand-drawn characters from 50 different alphabets with few examples per class. k-way N-shot classification means that a model is given a query from an unseen class and a support set S. S contains N examples each from k different classes and must recognize which set of examples the query sample belongs to.

Method	20-way 1-shot	20-way 5-shot
Pixels Nearest Neighbor	26.7%	42.6%
Convolutional Siamese Net [70]	88.0%	96.5%
Matching Network [147]	93.8%	98.5%
ConvNet with Memory Module [65]	**95.0%**	**98.6%**

Table 2.2.2 – Few-shot learning on Omniglot

Results on few-shot learning suggest that ConvNet with Memory Module pro-

vides promising results to learn from few examples, by defining examples designed to summarize classes.

2.2.6 Conclusion

Metric learning is a promising avenue of research, allowing classification of inputs into a large number of classes. Hyperbolic spaces in particular have garnered a lot of attention lately but require changes in gradient computations, and we did not manage to reproduce similar results on VRD.

Furthermore, VRD could benefit from few-shot learning approaches, as being able to learn from few examples is a useful feature for tasks where annotated examples are highly skewed.

However, for complex and high-dimensional inputs such as everyday life images, these techniques first require the extraction of an intermediate, lower-dimensional representation. Training models to extract representations that capture the differences between classes and generalize to new samples requires high amounts of data, not always available for studied benchmarks. Thus we focus in the next Section on how to train such models by using data from other sources, which we call External Data.

2.3 Learning with external data

As described in Section 2.1, most machine learning tasks can be formulated as finding a mapping from a vector space X to an output space Y that maximizes an objective function. However, this space is directly learned from the training data which is not always sufficient. This occurs in two cases. First, when the amount of available data is not enough to evaluate the objective function on a space similar to the real data distribution. Second, when the dimensions of the input and output space are high when compared to the size of the dataset. These conditions result in the difference between the experimental risk and the true risk being too high. In these cases, the model does not generalize well to new samples. The amount of data necessary to train the model depends on the dimension of both spaces, what objective function is used, how the model parameters are initialized, etc... In this section we study methods proposed to train a model with low amounts of data. For this, knowledge is integrated in neural networks through a variety of ways, from model initialization to training and inference.

2.3.1 Transfer Learning

Transfer learning refers to the training of a parametric model on a source dataset and task, then using part of the model weights in a new model to be trained on a target dataset, potentially for a different task. There exist several strategies concerning how the trained weights are updated during the second training, whether they are completely frozen or updated with a smaller learning rate. This has been shown to improve performance when the target dataset is too small to learn representations of the input that generalize well on the test set. In computer vision, the most commonly used source dataset used for transfer learning is ImageNet [49], a dataset with 1.2 million images and 1000 image classes.

Oquab et al. [106] show that training an AlexNet [72] network on Image Net [49] and using the weights of all but the last layers allows the model to learn to complete various tasks, such as object and action classification. Xu et al. [159] show that training on Flickr30k for image caption generation improves the BLEU score by 4 points on Flickr8k. However, transfer learning has limitations when the source and target datasets have very dissimilar data distributions. For example, training a caption generation model on the greater MS-COCO dataset [86] degrades the BLEU score on Flickr8k as the words are drawn from dissimilar vocabularies.

Hong et al. [61] explore transfer learning in the context of weakly-supervised learning. They use transfer learning in two different ways: the first to train an image encoder on ImageNet. For the second, with the encoded image as input, they train a two branch network: the first branch for image classification and the second

one to produce a segmentation mask. The target dataset is the Pascal VOC [32] which provides only image level labels. During training, the MS COCO dataset is used to train the segmentation branch on the same image representation as Pascal VOC, in the hope that it generalizes well on that same dataset. This provides a significant improvement when compared to weakly-supervized methods, and is comparable to semi-supervized settings, where the full annotations are available for only a subset of samples. Many recent works [17, 24, 43, 85, 88, 173] instead pre-train on MS-COCO [86] as it provides consistent improvement on the scene parsing task, as shown in Table 2.3.1. The Intersection over Union (IoU) metric is defined, for a bounding box prediction and its related ground truth boxe, as the ratio of areas of the intersection and union of both boxes.

Method	Description	Mean IoU
TransferNet [61]	Weakly-supervized	51.2
No Pretraining	Trained on Pascal VOC [32]	70.8
Mid-Level representations [106]	Pre-trained on ImageNet [49]	82.8
Pyramid Network [173]	Pre-trained on ImageNet [49]	82.6
Pyramid Network [173]	Pre-trained on MS-COCO [86]	85.4

Table 2.3.1 – Performance of Transfer Learning on Pascal VOC 2012 [32]

Since our task shares similar aspects to the scene parsing task we pre-train networks on MS-COCO [86].

2.3.2 Multimodal Learning

The previous section showed how annotated images and data from a similar domain can be used to train a classifier on pre-trained representations that generalize well and can compensate lack of data on the target class. Here, we explore a similar method, that makes use of data on the same domain but on a different modality. These methods are especially useful when the target domain is far from more studied domains but has easily available data in other modalities, such as image text, sound, etc...

Coordinated Representations Multimodal learning is the process of jointly learning the representation of data from different modalities. This can alleviate the difficulty of learning how to represent classes with a low number of training examples. Baltrušaitis *et al.* [7] show many of the challenges that this poses as different modalities have different levels of noise and examples from different modalities do not always match. Multimodal data can be combined several ways, but we

focus on the separate representation of data from different modalities, coordinating them through constraints, called coordinated representations.

In DeViSE by Frome *et al.* [38], text and image data are abundant but paired text-image data is scarced. Thus DeViSE is first pre-trained to learn visual and textual representations separately. Second, for examples with paired textual and visual data, a similarity constraint is enforced between the two modalities, as in 2.2. Socher et al. [133] enforce similarity constraints between paired captions and images. Captions are processed by a dependency tree RNN which extracts a representation where words contribute based on their position in the dependency tree.

Structured coordinated representations are learnt with additional constraints on the multimodal space. To make use of existing hierarchical relations between concepts, Vendrov *et al.* [144] enforce order relations between hypernym, concept, compound concepts and images corresponding to these concepts. This allows them to predict entailment between sentences, and to rank captions corresponding to new images and vice versa, increasing image recall@1 from 31% to 38% on MS-COCO [86]. Zhao *et al.* [172] enable open vocabulary learning by enforcing the same order constraints on hypernyms and the visual features of detected objects and show that hierarchical metrics (Wu-Palmer similarity) show more semantically consistent predictions. The F-score metric is improved from a softmax-based for Zero-shot scene parsing from 0.53 to 0.62.

Method	Description
DeViSE [38]	Similarity constraint between text and image
GCS [133]	Similarity between sentence root node and image
Order relations [144]	Order constraints on compound concepts and images
Open Vocabulary [172]	Order constraints on object features and concept hypernyms

Table 2.3.2 – Multimodal learning

Table 2.3.2 summarizes these methods. Multimodal learning requires paired data which is not always available, and the different levels of noise make the exploitation of both challenging.

2.3.3 Knowledge Graphs

Many sources of external knowledge such as semantic networks, synonym dictionaries, knowledge bases, contain valuable information that can complement datasets. In order to increase the generalization power of concept representations extracted by the network, an additional loss is added to the training objective of the network.

This loss quantifies to what degree the output of the network satisfies constraints based on the external knowledge. Thus Deng et al. [26] define a hierarchy and exclusion (HEX) graph, and integrate this knowledge into their model by defining its output as a Conditional Random Field. They show that it can be applied to diverse domains such as scenes, objects, actions... For object detection, knowledge graphs (KG) bring additional information in the form of relations between object classes. They can be integrated to neural networks to produce detections consistent with these relations. For this, information is shared among nodes of the graph using LSTM networks [95] or Graph Convolutional Networks (GCN) [150], or estimating the relatedness of objects with a random walk on the graph [33].

Table 2.3.3 shows the methods advantages and drawbacks. In the context of visual relation detection, knowledge graph can be straight-forwardly extracted from datasets and provide common-sense knowledge. HEX-Graphs [26] are not adapted for this use case but the three other methods can be used to improve semantic consistency among predictions on each image.

Method	Advantage	Drawbacks
HEX Graph [26]	Diverse application domains	No probabilistic relations
KG-GCNet [33]	Adapted to large-scale knowledge graphs	Low improvements on simple scenes
LSTMs [95]	Improved detection of small objects	Highly dependent on KG quality
GCN [150]	Robust to noise in KG	Information Dilution

Table 2.3.3 – Knowledge Distillation

Knowledge graphs garner increasing attention but they are hard to exploit, as they need to be well crafted or used with very robust models. In the context of image processing, it is disambiguating textual concepts related to visual concepts in semantic networks (*e.g.* disambiguating "*arm*" between a weapon and a body part) such as WordNet [34] or ConceptNet [137] is an avenue of research itself. In order to use more readily available data, we did not focus our study on this particular source of data.

2.3.4 Hierarchical Semantic Modelling

In Section 2.3.2, several methods defining a space in which visual concepts are represented have shown that enforcing constraints between semantic concepts helps the model learn representations of unknown concepts consistent with their place in the hierarchy of concepts.

Deng *et al.* [25] note that there exists a trade-off between predicting object labels with high accuracy and high specificity, and that a model should balance between both in case of uncertainty. They define an algorithm that maximizes the information gained (by predicting lower-level labels in the WordNet [34] hierarchy) while maintaining an arbitrarily high accuracy.

Ordonez *et al.* [107] build on [25] and combine image content estimation with a "naturalness" function that measures a trade-off between selection of concepts on varying levels in the WordNet [34] hierarchy and their n-Grams frequency in the Google 1T corpus [12].

Finally, Zhu *et al.* [175] augment WordNet with affordances (actions that can be performed on objects) and object attributes such as weight, size, color, texture... They learn a knowledge base (KB) using a Markov Logic Network and use the knowledge base to predict affordances on new objects.

Method	Description
DARTS [25]	Trade-off between accuracy and information gain
NaturalNess [107]	With word frequency
Affordances [175]	Learn KB with attributes and Markov Logic Network

Table 2.3.4 – Hierarchical Semantic Modelling

The three methods summarized in Table 2.3.4 use external data to adapt predictions to their frequency and likelihood in the additional data. Thus they are well adapted when the data distribution of the training set is either hard to estimate or is very different from the expected test distribution.

2.3.5 Conclusion

Transfer learning is a well established approach to extract representations of images or regions that can be used in different datasets, allowing the input space of downstream networks to be significantly smaller, dividing by 100 the space dimension. Multimodal learning, knowledge graphs and hierarchical semantic modelling suffer from the differences in distributions and levels of noise between the textual and visual modalities. They require methods that are robust to this noise. However, many presented methods show promising results, especially in the context of zero-shot learning.

For VRD-set and Visual Genome [71], relation boundaries are fuzzy. Additionally to the ambgiuity of classes of objects, we focus not on binary relationships between concepts but instead on more fuzzy data, such as similarity relationships.

Approaches presented in this section targeted limitations that models encounter when Internal Data, *i.e.* data from the target training set, is insufficient to accu-

rately estimate the true risk. We showed that this can be targeted by introducing knowledge into the network. However, this sometimes also stems from the fact that the model is not able to capture the whole signal from the training set, due to variance, noise, architecture limitations, and so on. For this reason, in the next section, we present approaches aimed at better capturing this signal from the training data itself.

2.4 Learning with internal data

In many cases, the model under study is not able to learn for the target task, due to too high variance, noise or too little data. However, domain knowledge can help devise new features, examples, from the already available data in order to decrease the parameter space or to improve the estimation of the objective function..

2.4.1 Data augmentation

The purpose of data augmentation is to use knowledge of invariants that a given model should preserve. This knowledge is used to generate a set of examples $\{(x'_1, y'_1), \ldots, (x'_n, y'_n)\}$ for each sample (x, y). Let a mapping f^* defined as in Equation 2.1. The goal is thus to define a new dataset $\mathcal{D}'$ such that

$$|\mathbb{E}_{(x,y)\sim p}[\phi(f(x), y)] - \tilde{R}(\mathcal{D}')| < |\mathbb{E}_{(x,y)\sim p}[\phi(f(x), y)] - \tilde{R}(\mathcal{D})| \tag{2.11}$$

where $\tilde{R}(\mathcal{D}) = \frac{1}{n}\sum_{(x,y)\in\mathcal{D}} \phi(f(y), x)$ is the empirical risk on dataset $\mathcal{D}$.

In [72], patches and their horizontal reflexions are extracted from each image and used with the same label as the original image, which multiplize the size of the dataset by 2048. Furthermore, they alter the pixel values by adding values based on the eigenvectors and eigenvalues of the covariance matrix of the R,G and B variables and a random variable drawn at each image. By this, they enforce the network outputs to be invariant to changes in intensity and color of the illumination. Dosovitskiy *et al.* [30] perturb images with cropping, affine transformations, rotation and contrast alterations and aim to classify the perturbed images as the same class as the original one.

2.4.2 Knowledge distillation

Knowledge distillation was introduced by Hinton *et al.* [59]. They note that ensemble models provide predictions with higher accuracy than single models. Indeed, in [28], Dietterich shows that training and aggregating the predictions of different models

— reduces the risk of choosing the wrong classifier

— provides a better approximation of the true unknown optimal function

— expands the space of representable functions

But these methods require to keep a set of models at test time which is computationally heavy in the case of neural networks. Instead, Hinton *et al.* propose to train a "student" model to reproduce the output of the ensemble method, called the teacher model. This output is thus considered as a soft label instead of the hard

labels that are used in conventional supervised settings.

Let $\mathbf{v} = \{v_1, \ldots, v_n\}$ the logits of an ensemble model, *i.e.* the aggregation of logits of several neural networks. The corresponding probability distribution is computed with the softmax function with temperature τ:

$$q_\tau(y = i|x) = \frac{\exp(\frac{v_i}{\tau})}{\sum_j \exp(\frac{v_j}{\tau})} \tag{2.12}$$

The student network is trained to minimize the weighted average:

$$\mathcal{L}(\theta, y|x) = \lambda l(q_\theta, y) + (1 - \lambda) l(q_\tau, y) \tag{2.13}$$

where q_θ is the student output distribution, l is the cross-entropy loss and λ is a weight chosen empirically. This method has been extended to the context of training a student model from a single teacher model. The underlying intuition is that the teacher network discovers relationships between the different classes that are used to increase the stability of training.

Gupta *et al.* [56] train a network on ImageNet and use it as teacher for a student network with image depth input on paired RGB/depth images. Li *et al.* [81] train a teacher network on a small and clean dataset, with confidence propagated through a knowledge graph and train the student model on a dataset with noisy labels. Radosavovic *et al.* [116] train an ensemble model where each simple model is trained from scaled and flipped data and train the student network on unlabeled data with the output of the ensemble model. Yim *et al.* [163] train the student model to reproduce aggregations of intermediate feature maps of the teacher model instead of its final output. The student model is shown to outperform and converge faster than models trained with transfer methods. Chen *et al.* [16] train a student network for object detection by distilling knowledge both for bounding box regression and object classification, weighing object and background classes with different weights and bounding the student regression by the teacher regression output. Furthermore, they show that additionally distilling intermediate features maps helps reduce under-fitting, improving training and testing accuracy. On Pascal [32], the distillation of a VGG16 network increase the mean Average Precision from 59.8 to 63.7.

Methods presented in Table 2.4.1 show applications of knowledge distillation on various tasks, showing that this technique is useful in many contexts and allows the model to better capture relations between features. Furthermore, they also allow smaller models to be trained and thus decrease computation costs.

Method	Description
Supervision transfer [56]	Student and teacher in different modalities
Noisy Labels [81]	Teacher trained on small clean dataset
Omni-Supervised Learning [116]	Ensemble of models on transformed data
Feature Map Distillation [163]	Aggregate and distill intermediate feature maps
Object detection [16]	Distill regression and multi-class classification

Table 2.4.1 – Knowledge distillation

2.4.3 Rule distillation

Rule distillation, introduced by Hu *et al.* [62], is similar to knowledge distillation, aiming at improving model generalization. The distinguishing factor from knowledge distillation is the use of additional knowledge of the domain under study, especially in the case of limited annotated data. Specifically, they show that neural networks can be trained to respect manually defined constraints. These constraints can come from external knowledge, that will supplement the available training data and make the model follow a set of rules that are difficult to learn from a low amount of data.

Let us consider a neural network with parameters θ that outputs a conditional probability distribution $p_\theta(y|x)$ of output $y \in \mathcal{Y}$ given input $x \in X$. Hu *et al.* [62] define a corrected probability distribution function $q(y|x)$ as

$$q = \arg\min_{q_{y|x} \in \mathcal{P}} \mathrm{KL}(q_{y|x}||p_\theta) - \lambda \mathbb{E}_{q_{y|x}}(\log f(x, y)) \tag{2.14}$$

where $\mathcal{P}$ is the set of conditional probability distributions. Thus q is defined as the projection of p_θ on a subspace verifying constraints defined by a function $f : (X, Y) \mapsto [0, 1]$. For $(\tilde{x}, \tilde{y}) \in X \times Y$, the more $(\tilde{x}, \tilde{y})$ respect these constraints, the closer $f(\tilde{x}, \tilde{y})$ is to 1. $\mathrm{KL}(q||p_\theta)$ is the Kullback-Leibler divergence from p_θ to q and $\mathbb{E}_{q_{y|x}}(f(x, y))$ is the expectation of $f(x, y)$ when the conditional probability distribution of $y|x$ $q_{y|x}$.

In the same vein as Knowledge Distillation, the objective function is a weighted average of the cross-entropy with the ground truth and the cross-entropy with the new corrected (teacher) distribution. Hu *et al.* [62] show significant improvement on several text-related tasks when compared to other regularization methods by enforcing simple logic rules on the model output. For example, on sentiment classification, the accuracy increase from 87.2% with a CNN to 89.3% with the same network trained with only one logical rule. Furthermore, contrary to other methods such as data augmentation, this method is able to adapt to the distribution of the data under study, as it takes into account the difference between the target distribution and the output distribution.

2.4.4 Self-supervised learning

In the context where available annotated data is not sufficient to train a model in a fully-supervized way, there exist methods that build upon the unannotated data and produce a training set. This makes it possible for a model to learn representations of data and extract pattern useful for downstream tasks. The important difference between this and data augmentation stems from the training of the model which is defined for task different from the target task whereas data augmentation adds additional data to training on the target task.

Doersch *et al.* [29] train a model to predict the relative positions of two sampled patches from the same image. By pretraining a model on this task with a relatively small dataset then training it on that same dataset for the target task (object detection), they improve the quality of learnt representations and of detections. Zhang *et al.* [171] use decolorized images and train a network to predict the two color channels. Pathak *et al.* [109] pre-train a model to predict the appearance of a patch removed from the image. Finally, Giradis *et al.* [44] show that training a model to recognize the rotation that has been applied to an image forces it to learn semantic features that improve its ability to perform various tasks such as object classification, detection and segmentation in an unsupervized setting.

Self-supervision methods can also learn useful representations in videos by predicting if a video is in the correct order [98], finding the shuffled sequence [36], predicting whether a video is played backward or forward [152].

Sentences describing images can also provide self-supervision. Modelling topics of an image has been proposed as a way to add self-supervision and improve the performance on various vision tasks. Mao *et al.* [93] infer image topics from captions with Latent Dirichlet Annotation [10]. They train a model to predict a latent topic from visual features and use it as a prior for image caption generation. For object detection, the same self-supervision from image topics has been shown to improve image classification, when compared to other self-supervized methods [52].

Similarly to knowledge distillation, Table 2.4.2 shows that self-supervision is helpful in many different contexts. The additional task constrains the representations learnt by the network, as they need to preserve information that does not directly correlate with data from the target task. Thus, without need for additional data, the network is able to better generalize. However these methods provide lower improvements than other transfer techniques such as transfer learning. For example, topic self-supervised models from Gomez *et al.* [52] reach 55.4% mean Average Precision on Pascal VOC 07 [32] where a model pre-trained on ImageNet reaches 73.6%. Results with self-supervision and transfer learning are not reported and

Method	Description
Context Prediction [29]	Relative positions of image patches
Colorization [171]	Colorize images
Context Inpainting [109]	Appearance of missing patch
RotNet [46]	Rotation of images
Shuffle and learn [98]	Verify frame order
Odd-One-Out [36]	Find out-of-order sequence
Arrow of Time [152]	Playing forward or backward
TextTopicNet [52]	LDA for Image classification
Caption Generation [93]	LDA Topics as priors

Table 2.4.2 – Self-supervised learning

further research is necessary to validate topic self-supervising.

2.4.5 Attention

In order to correctly predict an output for a given input, some parts of the input may hold less relevance than others. For example in the context of image captioning, only a fraction of the image may be relevant to predict the next word in the caption. This method is a form of self-supervision whereby the model learns to predict what parts of the input are relevant to the current context. Thus the attention mechanism makes the assumption that elements of the input should have a greater contribution in the final output of the mapping. Given a query or context vector q, the attention mechanism will attribute different weights to each element, quantifying to which extent each element is relevant to the query or context. The different types of attention mechanisms can be divided based on what those elements are:

1. Instance attention: when a mapping takes several vectors $(x_1, \ldots, x_n)$ as input, the attention mechanism attributes a different weight to each vector. For $i \in [1, n]$, $w_i = att(x_i, q)$. The mapping function can then be reformulated as:
$$f : (x_1, \ldots, x_n) \rightarrow f'\big(\sum_{i \in \{1 \ldots n\}} w_i \cdot x_i\big) \tag{2.15}$$

2. Feature attention: in the case of a single input, a weight is attributed to each dimension of the input. Let an input $x = (z_1, \ldots, z_p)$, for each $i \in \{1 \ldots p\}$, let $w_i = att(x, z_k, q)$, then f can be written as
$$f : x \rightarrow f'\big((w_1 z_1, \ldots, w_p z_p)\big) \tag{2.16}$$

The attention weights usually come from the output of a fully connected layer taking

as input the vector to be weighed. This output is passed into a softmax layer to produce a probability mass over each vector (*soft* attention). In order to reduce noise however, it is sometimes beneficial to threshold these scores to obtain a sparse vector (*hard* attention).

The attention mechanism was introduced first in [5] in the context of a Neural Machine Translator (NMT), which encodes an entire sentence in a single context vector. Before that, when dealing with long inputs, the context vector was inadequate to capture all the information required to translate the sentence. Allowing the model to search for relevant parts for the prediction of a word eases the information bottleneck at the context vector. Similarly, in image processing, attention allows to capture details in an image in order to generate captions [159]. More recently, Vaswani *et al.* [143] have shown that recurrent layers can be entirely replaced by a multi-headed self-attention.

In classification tasks, feature attention allows the model to focus on specific regions of the image as shown by Girdhar and Ramanan [46], improving the accuracy of an action recognition model. Specifically, each action class is considered as a different query. They combine bottom-up attention, conditioned only by the image features and top-down attention, conditioned by each action class. They obtain the score for each action by globally pooling the image representation $f(I)$ multiplied by the attention weights:

$$s_i = \sum_{0 \leq x \leq W} \sum_{0 \leq y \leq H} a_{xy} f(I)_{xy} \tag{2.17}$$

Method	Description
NMT [5]	Attention over word vectors
Multi-head Self-Attention [143]	NMT with Multi-Head Attention
Hard-Attention [159]	Hard attention over image locations
Attentional Pooling [46]	Product of top-down and bottom-up attention

Table 2.4.3 – Attention

2.4.6 Conclusion

Data Augmentation is the most widely used approach among the previous methods, especially due to the easy implementation, as they usually involve adding several samples for each training sample. Several data augmentations can be used together, making the model preserve several invariants. Knowledge distillation requires two consecutive training but is straightforward. However, we do not aim at reducing

model size thus will focus instead on other methods aimed at increasing generalization of the model without external data, such as rule distillation and self-supervised learning. The attention mechanism is very useful in order to aggregate inputs from different sources, such as objects, words, image regions... and improves model explainability, hence its wide use, especially in Natural Language tasks.

Finally, for both external and internal data, if the training data is very different from the test distribution, the trained model risks not generalizing. Indeed, in such cases, the evaluation of the loss function is biased and its optimization will not yield good results on the test set. Thus, after studying how best to learn to model input samples with external and internal data, we study how to learn from biased datasets.

2.5 Learning from biased Datasets

Highly skewed datasets, comprised of a high number of a small set of classes, can be a hurdle, preventing the model from correctly classifying instances into the less frequent classes. Furthermore, tasks that require matching pairs of examples have a very high number of negative matches and a naive selection is not enough to learn discriminant features. Buda *et al.* [13] divide these methods as follows.

2.5.1 Resampling

Skewed class distributions in training data have adverse effects on the learning process, as the sum of errors on rarer (minority) classes is far outweighed by that of the most frequent (majority) ones, and in the case of overlapping distributions, the trained model becomes biased towards the majority classes. Resampling consists of rebalancing the class distribution. Resampling methods fall into three groups:

— Over-sampling: creating new examples from minority classes.
— Under-sampling: discarding examples of the majority classes.
— Hybrid resampling is the combination of both methods

Over-sampling Synthetic Minority Over-sampling Technique (SMOTE) [15] and Adaptive Synthetic (ADASYN) [57] generate new samples of the minority classes by interpolation. ADASYN focuses on examples that are misclassified by a K-Means algorithm while SMOTE uniformly weighs examples.

Under-sampling Random under-sampling removes samples from the majority classes at random. Tahir *et al.* [139] propose to create a set of subsets of the training data where the majority class is severely undersampled, to the point of becoming the minority, called Inverse Random Under Sampling (IRUS). Training one classifier on each set, they use ensembling techniques to combine their votes.

Method	Description
SMOTE [15]	Example generation by interpolation
ADASYN [57]	Weighted example generation by Interpolation
IRUS [139]	Minority to Majority + Ensembling

Table 2.5.1 – Resampling

2.5.2 Example selection and weighing

To train SVMs, Hard Example Mining was introduced as a two-step process, alternating between updating the model parameters and updating the set of "hard"

examples. These hard examples are the examples that do not respect the margin constraint and are the only ones that contribute to the loss function [35]. Simo-Serra *et al.* [130] select hard negatives for a given pair of matching patches: non-matching patches with the highest loss. Wang and Gupta [149] select the K examples with the highest loss when training a model to match patches of a video representing the same entity. Shrivastava *et al.* [128] introduce Online Hard Example Mining (OHEM) and train an object detection model with two passes through the network: first, each proposed ROI is passed through the network in order to compute the associated loss and regions with the highest loss are added to the current mini-batch.

Lin et al. [87] propose a focal cross-entropy loss, changing the cross-entropy loss. Let p be the estimate probability that the class y associated to an input is equal to 1. The cross-entropy loss is computed as $CE(p_{gt}) = -\log(p_{gt})$ where $p_{gt} = p$ if $y = 1$ and $p_{gt} = 1 - p$ otherwise. The proposed focal loss is defined as $FL(p_{gt}) = -(1 - p_{gt})^{\gamma} \log(p_{gt})$. This increases the weight of examples that the model classifies with a lower confidence and decreases the weight of examples classified with a high confidence, especially negative examples, which in the case of tasks such as object detection, account for the majority of examples. They observe higher accuracy increase than Online Hard Example Mining, increasing the average precision from 31.8 to 36.0 on COCO [86].

2.5.3 Conclusion

Resampling and example selection are two widely used methods aimed at improving the estimation of the objective function. They are useful when this estimation would be biased with the original distribution of training data. In the case of existing VRD benchmarks [71, 82], since the train and test sets are drawn from the same distribution, this does not provide improvements on the test set. On the other hand, due to the very low proportion of positive object pairs - *i.e.* object pairs with at least one positive relation - detecting positives is a hard task. Example selection is a promising avenue toward learning what separates positive and negative object pairs.

The three previous Sections have shown how to better evaluate the objective function, using External and Internal data or artificially modifying the training distribution. Some of these methods have been applied to the case of Visual Relation Detection. We study in the next Section how they impact VRD models and how the issues we raised can be targeted.

2.6 Visual Relation Detection

In this section, we study existing methods proposed for VRD and show how they improve models designed for this task. Then we compare them with other existing methods and, after proposing several criteria, select what methods to explore in this work.

2.6.1 Standard Visual Relationship Detection architectures

Most standard Visual Relation models [23, 58, 102, 158, 161, 164, 165, 166] execute the following steps as shown in Figure 2.6.1.

1. Detect objects in the image. Using a Faster R-CNN model [120] (see Section 2.1.4) object detections are extracted in the form of bounding boxes.
2. Extract representations of both objects and relation using ROI-pooling [120] (see Section 2.1.4). The region corresponding to the relation is the minimum bounding box around the pair of objects.
3. Classify head and tail feature maps into object classes and relation feature maps into relation classes. head and tail may have different feed-forward layers or share the weights.
4. Generate the image scene graph by taking the top k scoring triplets, where each score is the product of class probabilities.

In the following chapter, works proposing variations from the described steps, especially step 4., will be studied in light of the issues associated with the task of Visual Relation Detection as stated in Section 1.2 and below.

As has been studied by Zellers *et al.* [166], with the existing benchmark datasets, the correct relation class is relatively easy to predict given a set pair of objects. Indeed, given the correct head and tail classes, the correct relation can be determined with a 97% accuracy by predicting the most frequent relations for this pair in under

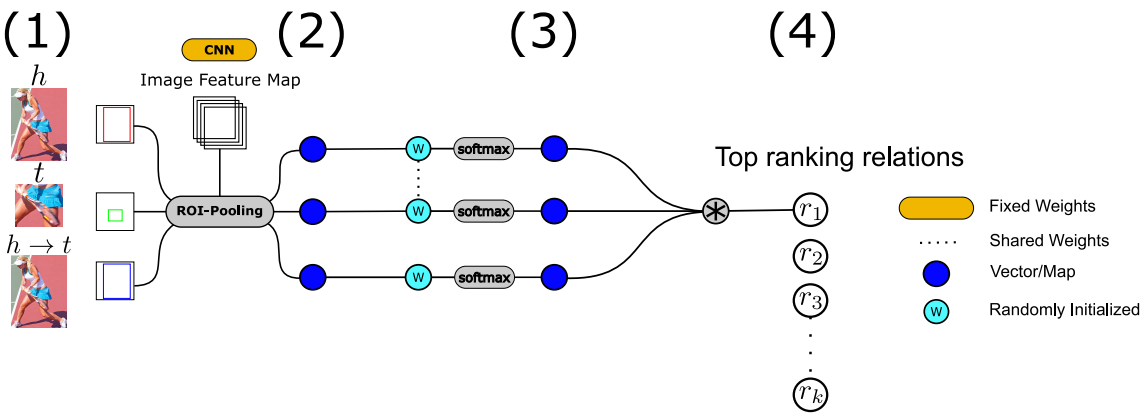


Figure 2.6.1 – Common VRD pipeline.

5 guesses. However, with an average of 21 objects per image in the Visual Genome, the detection of related object pairs is of paramount importance. Thus we study both aspects of Visual Relation Detection separately, first the classification of an object pair into a set of true relations; second, the estimation of the relevance of the object pair, which is a less studied aspect of VRD, *i.e.* its ability to detect whether two objects are related and how relevant these relations are.

2.6.2 Relation Separability and Classification

Visual relations have a high intra-class appearance variability as their appearance is highly dependent on the objects involved. For example the relation 'sit' will involve a person with legs in a certain position but the legs of a sitting cat may not be visible. Similarly, visual relations show a high degree of inter-class similarity as two different relations with similar pairs of objects will share a lot of common features, making it hard to learn the discriminant ones, as illustrated Figure 2.6.2.

(a) Cow standing on grass (b) Cow eating grass

Figure 2.6.2 – Examples from Visual Genome. Visual representations of relations are highly dependent on involved objects and different relations may be very visually similar.

Low level dependencies The appearance of objects is affected by the relations between them. For example the relation class "*sit on*" conditions the appearance of a person sitting on a sofa, the pose of this person. Based on this observation, several works propose to model these dependencies.

Information is passed from relation elements at different levels of abstraction of object and relation representations. First Dai *et al.* [23] iteratively refine head, tail and relation representation using feed-forward layers connecting each element of the relation. Li *et al.* [80] pass messages between elements at different points of the network, between feature maps before ROI-pooling (see Section 2.1.1) and between

Figure 2.6.3 – The appearance of legs is conditioned by the relation class "*sit*", but they may be folded for a human and not visible for a cat.

high-level representations after ROI-pooling. Another differentiating factor of this work is the fact that the feature maps before ROI-pooling are different between head, tail and relation because a different convolutional network is used for each. Yin *et al.* [164] directly pair information from relation elements with a spatially-aware context module. For relation features, each element is de-ROI pooled into a grid of fixed size representing the whole relation. Thus the head and tail features are equal to zero outside of their bounding box. Then for each (head, tail), (head, relation), (tail, relation) pair, channels from both elements are fused before being passed into the following convolutional layers at two points of the network. Thus the model is able to learn spatial relationships between features of the head, tail and relations.

Xu *et al.* [158] take a similar approach, called Iterative Message Passing (IMP), using a message passing framework. Instead of considering each triplet separately, information is propagated through the whole scene graph at each iteration, from nodes (objects) to edges (relations) and from edges to nodes (in the dual scene graph). Thus information about one pair of objects can influence the prediction of other object classes and relations. For example, the presence of a horse and a barrier can be correlated to the presence of a person riding the horse.

Newell and Deng [102] learn these interdepencies by learning to iteratively refine feature maps taking the whole context into account. A stacked hourglass network [103] learns to combine local and global information and reason over the full image. Two heatmaps are output by the model: one quantifying the likelihood of presence of an object and the other the presence of a relation. Objects and relations are decoded at the most probable locations. They improve especially the recall on smaller scene graphs, reaching state of the art performance on Relation Classification Recall@50 at 82.0%, increasing by 83% when compared to IMP [158].

High level dependencies Here we study methods which directly model interpendencies between relation elements at a high level, *i.e.* at the concept level, contrary to the previous methods which learn how lower-level features of each element influence the high-level concepts.

Liang *et al.* [84] train an agent to iteratively produce nodes and edges for the scene graph of an image with reinforcement learning. At each time step, the agent chooses to predict either an object, relation between two objects or an attribute. They are chosen among the set of relations and attributes that have been encountered in the training set for the current object. Thus inter-dependencies between relation elements are enforced directly at inference time instead of learned by the network.

Yu *et al.* [165] use the rule distillation method from [62] described in Section 2.4.3. During the training phase, the network learns to predict high scores for relations which are likely for a given pair of objects by integrating a constraint that penalizes low probabilities. On VRD-set, knowledge distillation increases the Recall@100 on Relationship Classification from 84.9% to 87.0% on the base network, to 89.4% on the corrected distribution and 94.7% on the combination of both.

Liang *et al.* [83] introduce a structural ranking framework, with a triplet loss aiming to increase the score difference between positive triplets (relations that hold true) and negative triplets. They adopt an adaptive margin for this loss, which increases the more likely the positive relation class is and the less likely the negative one is. Thus, missing or incorrectly predicting frequent relations are more penalized. Similarly, Sarullo and Mu [124] weigh the loss of relation mis-classifications using the relative frequencies of the two classes. For relation classes j and $k \in \mathcal{C}$, they define the cost of misclassifying an example of class j into class k by

$$w_{jk} = (1 - \delta_{jk}) \max(1, \log \frac{N_k}{N_j}) \tag{2.18}$$

where N_k and N_j are the number of examples of each class and $\delta_{jk} = 1$ if $j = k$ and 0 otherwise. They show that this increases the number of detections of uncommon classes. On VRD-set, this translates to an increase on Relationship Classification from 90.6% with a softmax loss to 93.2% with a structural ranking loss.

Hybrid In contrast to the above works, where high-level dependencies are used during training as an additional constraint that the mapping must respect while minimizing an objective function, Herzig *et al.* [58] use high-level information as input to the network in the same vein of low-level dependencies modelling. Indeed, the output of a pre-trained Visual Relation Detection is used as input to the new model, namely the probability distributions for each object and relation in the image. This

departs from the previous works where the input was the raw image. Furthermore, using relation predictions from a pre-trained model enables the model to add a pre-trained embedding of the most probable relation. This approach mostly focuses on predicting relations and objects with uncertain object detections. Thus, they reach state of the art performance on the Scene Graph Classification task reaching a recall of 50.8% (from 46.6% [166]).

Similarly, Wang *et al.* [148] use a memory module that iteratively refines object and relation predictions. For this, they combine probability distributions with visual features by adding the outputs of convolution and feed-forward layers. This information is then stored into a memory layer and updated using GRU cells [18]. It is then passed through an inverse ROI-pooling function in order to put it back into its spatial position and used as the visual feature for the next update. They show that the association among relations is important and learning which relations are often associated enhances recall. However, the added memory is mostly helpful to retrieve the more common relations as it is necessary to have enough examples in order to extract similar patterns. This improves their baseline on Relation Classification with graph constraints from 48.9% to 57.9% Recall@100, below state of the art methods at 71.0%.

Increasing separability with additional information Visual relations are very similar when similar objects are involved. Thus additional features have been proposed to increase their separability.

Spatial Features As shown in Figure 2.6.4, a large number of relations are spatial relations, such as 'on', 'behind', 'near'... Thus, adding spatial features that represent the spatial configuration of the object pair is a natural step towards learning separable relation representations. Peyre *et al.* [112] propose to encode the box size ratios, translations, aspect size ratios and overlap:

$$f_{spat}(o_1, o_2) = \left[\frac{x_2 - x_1}{\sqrt{w_1 h_1}}, \frac{y_2 - y_1}{\sqrt{w_1 h_1}}, \frac{\sqrt{w_2 h_2}}{\sqrt{w_1 h_1}}, \frac{o_1 \cap o_2}{o_1 \cup o_2}, \frac{w_1}{h_1}, \frac{w_2}{h_2}\right] \tag{2.19}$$

where x_j, y_j, w_i, h_i are respectively the x and y coordinates, width and height of object j, $o_1 \cap o_2$ $o_1 \cup o_2$ are the areas of the intersection and unions of the bounding boxes of o_1 and o_2. Furthermore, to take into account the multimodal nature of some relations such as 'next to', this vector is discretized into bins. For this, the spatial configuration is summed to be generated from a mixture of k Gaussian and a Gaussian Mixture Model is fit to the training pairs.

Several works [23, 166] adopt an end-to-end approach by training the network to learn directly the representation of the spatial configuration. They instead extract

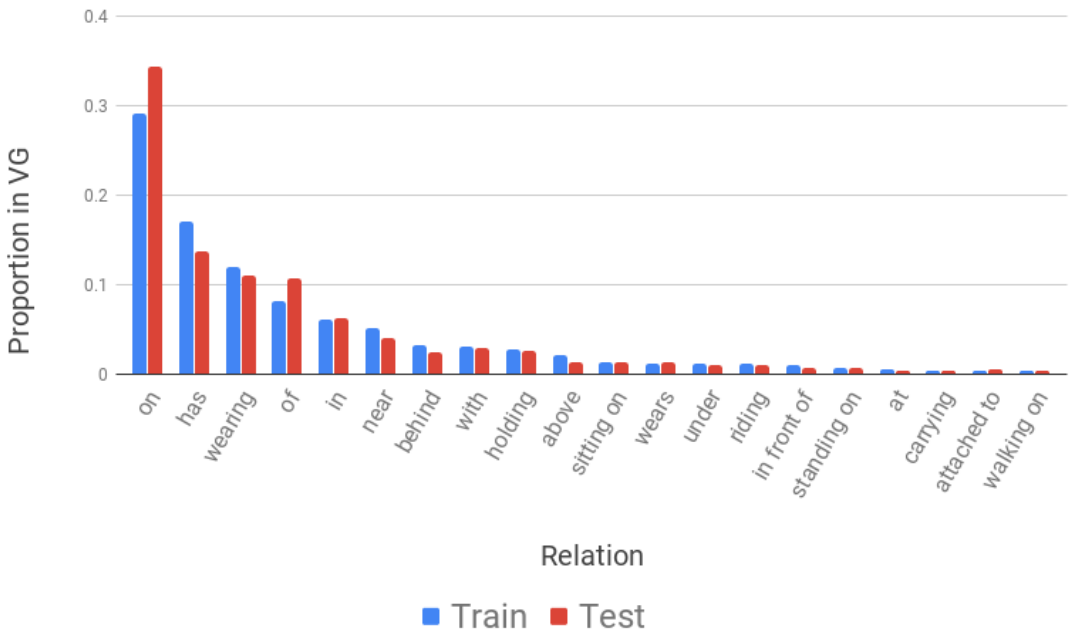


Figure 2.6.4 – Relation distribution in Visual Genome. Spatial relations represent 58% of the annotated relations.

a binary mask for each object, where the image is represented by a grid of fixed size. At each location (i, j) of the grid, the value is 1 if (i, j) is inside the object bounding box and 0 otherwise. Then, the masks of both head and tail are considered as channels of an image and passed into a convolutional network to extract the representation of the object pair.

Woo *et al.* [154] take an intermediate method and use learned spatial representations by plunging the relative location and scale information vector into a higher dimensional space:

$$f_{spat}(o_1, o_2) = \boldsymbol{W}\left(\frac{x_2 - x_1}{w_1}, \frac{y_2 - y_1}{h_1}, \log\frac{w_2}{w_1}, \log\frac{h_2}{h_1}\right) + b \tag{2.20}$$

where $\boldsymbol{W}$ and b are learned weights and biases.

Semantic features Many of the previous works [58, 166] use pre-trained word embeddings [97] concatenated to the visual representations of objects.

Contextual information In [58], contextual features are concatenated to the representation of each object. They are comprised of the number of larger and smaller objects in the images, the number of objects to the *left, right* and *above, below* as well as the number of objects with higher and lower confidence.

Learning Under-represented classes with Semantic Modelling Lu *et al.* [89] integrate language priors into a language ranking module, combining both visual and language modules at test time. For this, a mapping projects pairs of pre-trained word vectors into the relation space where each coordinate corresponds to a score for the given pair of object classes. This mapping is trained to optimize two objective functions. The first quantifies how different the distances are between relations in the pre-trained vector space and in the learnt space. The second encourages higher scores for more frequent relations. Learning to preserve distances from the pre-trained space helps the network learn from similar relations and improve visual relations detection.

In a similar fashion, Zhang *et al.* [169] integrate language priors into the visual space. They define two different multimodal space: one to represent objects and the other to represent relations. In the same vein as Kiros *et al.* [69], they use a triplet loss (see Section 2.2) to enforce visual representations of different object (resp. relation) classes to have a higher distance than a given margin. This enables the network to make use of language priors to learn a consistent visual representation and correctly classify examples of relations from the tail of the distribution (with fewer than 1024 examples), increasing the top10 accuracy from 7.7% to 28.1% on VG80k.

Yin *et al.* [164] adopt a hierarchical approach, defining two trees organizing both object and relation classes. At the lowest level, object and relation classes are taken as they are. On the first level, each object and relation class is filtered and normalized using a part-of-speech tagger and lemmatizer. For objects, the last level is added by clustering labels with a Leacock-Chodorow [79] distance under a set threshold. For relations, classes with the same preposition are clustered together in the 'spatial' partition while classes with the same verb are clustered together in the 'action' partition of the hierarchy. Thus for both objects and relations, the model is trained to output three probability distributions, one for each level of the hierarchy and the losses for each level are added.

2.6.3 Relation Detection and Relevance Classification

Supervised Relevance Classification The number of object pairs for each image is equal to $n(n-1)$ where n is the number of detected objects. Computing the representation of each object pair is computationally heavy. Thus Yang *et al.* [161] train a Relation Proposal Network that prunes the scene graph by learning a scoring function and taking the top K ranking edges. For this, the model learns two projections from the object representation to a vector space, one for the head of the relation, the other for the tail. The pair score is defined as the dot prod-

uct of those vectors. After the scene graph has been pruned, each remaining edge is classified using an attention mechanism propagating information between connected nodes. Similarly, Sarullo and Mu [124] learn a binary classifier that takes each object pair representation as input and outputs, using a sigmoid function, the probability that the relation is not a "background" relation. The resulting scene graph is then pruned from pairs where this probability is lower than 0.5. Results on most methods improve the F1 score on most methods but due to the incompleteness of the task, precision measurement are very inaccurate. Furthermore this approach deteriorates class macro recall, which is an important metric, as we argue in Chapter 3. Thus the impact of this method would require further study on more complete annotations.

Weakly-supervised Relation Detection With the observation that the combinatorial nature of the problem makes the annotation of images inherently incomplete, Peyre *et al.* [112] propose a weakly supervised model that learns to detect relations from image-level labels. For this, they make the hypothesis that each image-level relation annotation has a latent assignment to a pair of objects. Thus the model is training to both minimize the error of each relation classification with a given relation-object assignment and to minimize this error with respect to the assignments. They show that when all objects detections and their classes are given, the precision of relation classification only drop from 50.4% to 46.8%.

Self-supervised Relation Detection Woo *et al.* [154] integrate contextual information by relational embedding. Each object feature is defined as a concatenation of image and object visual features as well as object classes. They are then refined using an attention mechanism over all objects, called relational embedding. They show that this embedding coincides with ground truth relations and represents the inter-dependencies between objects. Relation probabilities are computed with after addition relational embedding, using one-hot object encodings, the output object representations and the union bounding boxe features. They reach state of the art results on most tasks with graph constraints, reaching 68.5% on Relation Classification from 67.1% of MotifNet.

2.6.4 Summary of Visual Relation Detection methods

The presented methods tackle issues related to the task of Visual Relation Detection, we summarize what differentiate them in Table 2.6.1. Before analysing their respective advantages and drawbacks, and proposing directions to tackle the identified issues, we present existing benchmarks available to evaluate the performance of Visual Relation Detection models.

Method	Description
Relation Classification	
Low-level dependencies	
DR-Net [23]	Message passing after ROI-pooling
ViP-CNN [80]	Message Passing before and after ROI-pooling
SCA-M [164]	De-ROI pooling for spatially consistent fusion
IMP [158]	Message Passing through dual Scene Graphs
Pixels2Graphs [102]	Local and global information through hour glass
High-level dependencies	
Variation-structured RL [84]	Enforce dependencies at interference
Rule Distillation [165]	Train model with high-level constraints
Structural Ranking [83]	Penalize frequent rel. for missing or false positives
Class Imbalance [124]	Weigh mis-classification with frequencies
Hybrid	
Mapping Images to SG [58]	High-level features input and message passing with neighbour attention
Additional Features	
Weakly-supervised [112]	Spatial features fit into GMM
CNN Spatial Features [23, 166]	Learnt representation from object binary masks
Spatial Embedding [154]	Plunge spatial features into high dimension space
Word embeddings [58, 166]	Concatenate word embedding representation
Context information [58]	Position of other objects and rankings
Semantic Modelling	
Language Priors [89]	Preserve dist. from pre-trained to trained relation space
Zoom-Net [164]	Predict classes at different hierarchy levels
Large-scale VRD [169]	Triplet loss in two multimodal spaces
Relation Detection	
Fully-supervised	
Triplet NMS [80]	Filter triplets with high overlap
Graph R-CNN [161]	Model relevance by dot product
Pixels2Graphs [102]	Relations detected at Heatmap maxima
Background filtering [124]	Remove edges under a threshold
Weakly-supervised	
Weakly-supervised [112]	Learn latent object-relation assignments
Self-supervised	
Relational Embedding [154]	Attention on object pairs quantifies relevance

Table 2.6.1 – Visual Relation Detection methods

2.7 Evaluation and experimental datasets

2.7.1 Datasets

Stanford 40 actions [162] is a dataset with 40 actions performed by humans. It contains 9,532 images with 180 to 300 images per action class. Actions can have an object, such as 'cutting_vegetable' or no object, such as 'applauding' as shown in Figure 2.7.1

(a) Applauding (b) Cutting vegetables

Figure 2.7.1 – Images from Stanford 40 actions dataset [162].

VRD-set [89] is comprised of 5000 images, split into 4000 train images and 1000 for test. Ground truth object bounding boxes and classes are provided along with relation annotations for a set of object pairs for each image, with around 35,000 annotated relationships with 100 object classes and 70 relation classes.

(a) *(cat, on, luggage)* (b) (person, hold, umbrella)

Figure 2.7.2 – Images from the VRD-set dataset [89] with one relation example.

Visual Genome [71] has 108,077 images. We focus on the split defined by Xu et al. [158], restricted to 150 object classes and 50 relation classes with an average of 22 relationships annotations per image. 70% of images are used for train and 30% for test.

There does not exist an official split for Visual Genome, thus several different splits have been proposed with different number of selected classes:

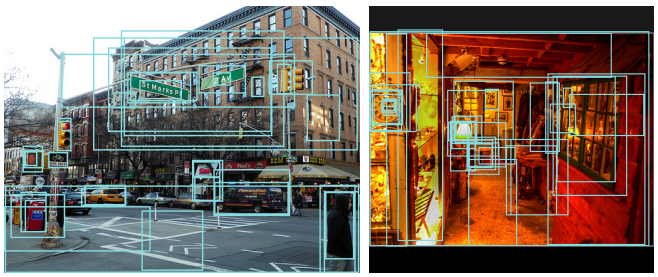


Figure 2.7.3 – Images from Visual Genome [71].

Dataset	Source	Train	Test	Object classes	Relationship classes
VG	[158]	73,800	25,800	150	50
VG	[167]	73,800	25,800	200	100
VG	[164]	86,460	21,600	5,319	1,957
VG	[169]	99,900	4,800	53,300	29,000

Table 2.7.1 – Characteristics of existing splits of Visual Genome

UnRel [112] To evaluate the ability of the model to generalize to and detect new relation triplets, Peyre *et al.* [112] generated a new dataset by querying a search engine. Then each retrieved image is annotated with object bounding boxes and relations the corresponding to each of the selected queries. Thus **UnRel** contains 1,000 images with 76 triplet queries.

OpenImages [74] has recently been augmented with visual relation annotation. It contains **9 million** images with 10 relation classes and 600 object categories.

HICO [14] has 47,000 images with 117 common human-object actions covering 600 action-object pairs such as (*ride, bike*) and 80 object categories. Thus the main difference with previous datasets is the exclusion of relations where the subject is not a human. Furthermore images are only annotated with image-level labels and no bounding boxes for objects. Thus images in the dataset are less cluttered with only one salient human-object pair. This makes the detection of relation less important and focuses mainly on their classification. Finally, by decreasing the number of pairs under consideration, annotations are more complete.

Actions in videos Many benchmarks such as HMDB [73], UCF101 [136], Moments in Time [100] and Kinetics [66] focus on action classification in videos with actions such as human-object and human-human interactions, body-motions or sports.

Conclusion We focus on visual relation detection on two datasets: VRD-set [89] and Visual Genome [71] as they are widely studied image datasets with annotated

object bounding boxes and relations. Furthermore, because of the high number and diversity of images and relations in Visual Genome, the evaluation of models on this benchmark gives accurate estimates on the performance of models on daily life images. Finally, Visual Genome has more cluttered images which provides a challenging problem due to the resulting high number of possible relations per image and incompleteness of annotations.

2.7.2 Evaluation tasks and metrics

Scene Graph Detection and Classification Annotations of visual relations are inherently incomplete. This is due to the combinatorial nature of the problem and to the large quantity of information in images, which makes it necessary for humans to focus on specific regions. The lack of comprehensive annotations negatively impacts the quality of the evaluation of VRD models and makes precision metrics too pessimistic. Hence most works on VRD [23, 58, 102, 158, 161, 164, 165, 166] measure the *recall@k*: the fraction of ground truth annotations among the top k candidates, as proposed by Lu *et al.* [89]. Setting a maximum number of relations per images allows to take the precision of the model into account and keep the number of false positives low.

This metric is used for the following tasks:

— **Relation Classification** (RelCls): ground truth object bounding boxes and classes are given and the model is evaluated on the quality of relation prediction.

— **Scene Graph Classification** (SGCls): only ground truth bounding boxes are given and we evaluate the quality of object and relation classification.

— **Scene Graph Generation** (SGGen): The model is evaluated on its performance on object detection and relation classification. An object detection is correct if the ratio of intersection area over union area (IoU) is a least 0.5.

However, both SGCls and SGGen heavily punish misclassification of objects or relations. Indeed, for each $(subject, relation, object)$ triplet, an error in the label of any of the three elements will result in a false positive. Thus, for scene graphs where a few nodes are connected to several others, an error in the classification of center nodes will result in missing all relations with these nodes. To remedy this while keeping the accent on object classification, contrary to RelCls, Yang *et al.* [161] propose to instead compute the combined recall of object, relation classes and relation triplets :

$$\text{SGGen+} = \frac{C(O) + C(R) + C(T)}{N} \tag{2.21}$$

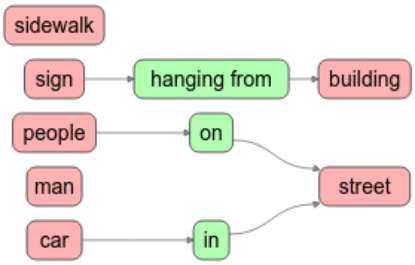
where $C(O)$ (resp. $C(R)$, $C(T)$) is the number of correctly classified objects (resp. relations, triplets) divided by the sum of annotations of each type. $C(T)$

Graph Constraints Finally, setting a number of predictions for each object pair balances the emphasis on the quality of relation detection or relation classification. Many works have focused on using 'graph constraints' which limits the model to one class prediction per object and per object pair. This focus ensures a high precision but severely limits the recall. Furthermore, it also decreases the emphasis on relation detection, as predicting one relation for the wrong object pair is less penalized than predicting several for that pair. On the other hand, several relations may be true for each object pair. For example, the relation '(cat, sleep on, bed)' implies '(cat, on, bed)' thus it makes sense to predict several for the pair '(cat, bed)'. However, for images with fewer objects or with fewer likely related object pairs, increasing the number of predictions per pair allows the model to predict several false positives without being penalized. Thus several works [102, 161, 164, 166] measure the previous metrics with k predictions per object pair with $k = 1, 10$ or 70.

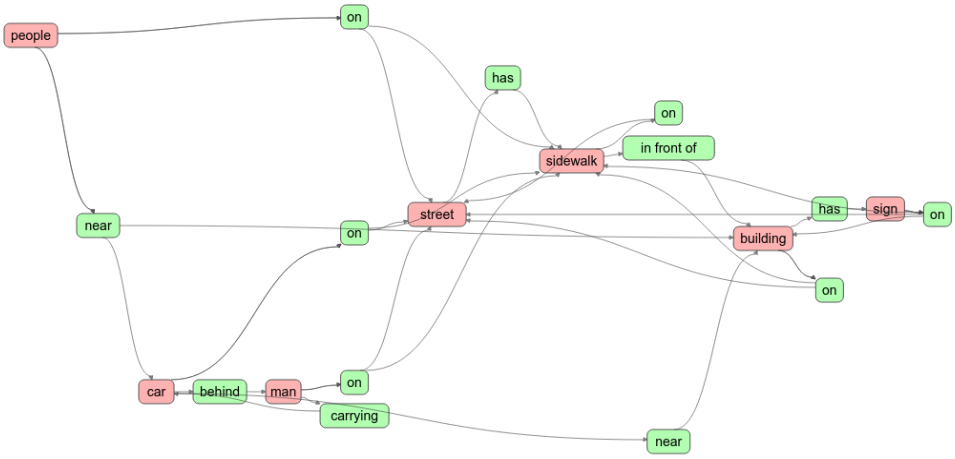
On Figure 2.7.4, outputs of our baseline are displayed with and without graph constraints. Without graph constraints, the number of annotated pairs decreases. While some relevant pairs might be missed, irrelevant pairs are also removed, allowing more true predictions on a single object pair, such as (*car, street*).

These observations lead to the conclusion that is important to allow the prediction of several relations per object pair, with a low number of predictions on the image, in order to keep a high precision.

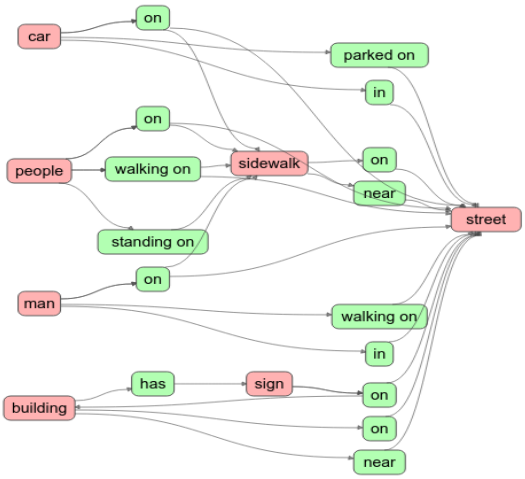
(a) Input image



(b) Ground truth



(c) Scene graph with graph constraints: at most one relation is predicted per object pair.



(d) Scene graph without graph constraints: the pair (*people*, *sidewalk*) has three true predicted relations.

Figure 2.7.4 – Image input with ground-truth annotations (*top*) and output annotations from our baseline (see Chapter 3) with (*middle*) and without (*bottom*) graph constraints.

Zero-Shot learning To evaluate the ability of the Visual Relation Detection models to generalize, several works [84, 89, 112] evaluate their performance on previously unseen '(subject, relation, object)' triplets, filtering out every seen triplet from the test set. This highlights the ability of the model to recognize relations in spite of uncommon paired objects, which helps show whether the model learnt spurrious correlations or relies too much on the object classes.

2.7.3 Impacts of Methods on results on VG and VRD-set

We report the impact of the proposed methods on the task of Visual Relation Detection on both Visual Genome and VRD-set datasets.

Method	RelCls
IMP [158]	72.6
DR-Net [23]	81.9
Rule Distillation [165]	81.9
Zoom-Net [164]	90.6
DSR [83]	**93.1**

Table 2.7.2 – Visual Relation Detection Results on VRD-set with no graph constraints

Method	SGCls	RelCls
VRD-set [89]	14.1	35.0
IMP [158]	24.4	53.0
Pixels2Graph [102]	22.6	55.4
NeuralMotifs [166]	36.5	**67.1**
SGP [58]	**38.8**	66.9
Memory [151]	29.5	57.9

Table 2.7.3 – Visual Relation Detection Results on VG with graph constraints

Method	SGCls	RelCls
IMP [158]	29.2	58.2
Graph R-CNN [161]	31.8	59.1
Pixels2Graph [102]	30.0	75.2
NeuralMotifs [166]	**47.7**	**88.3**

Table 2.7.4 – Visual Relation Detection Results on VG with no graph constraints

On Visual Genome, which most recent works focus on, Neural Motifs [166] is the best performing model for the relation classification task, thanks to an improved pre-processing and negative example selection, which they show improves the performance of several other methods and the addition of a recurrent model to take into account dependencies among relations of a single image.

2.8 Analysis

In Table 2.8.1 targeted criteria and issues of supervised learning in general and VRD in particular are shown for the previously described methods. All aspects and issues are formulated so that + refers to an advantage of the method and − is a drawback. Relation and Relevance Classification are defined in Section 1.3.

— Reduced impact of noise/variance in Class variable: ClsNoise
— Helps scaling to a large number of classes: Scale
— Reduced under-fitting without additional annotated data: Fit
— Less Susceptible to dataset Bias or Differences in Domain Data Distributions: Bias
— Less Susceptible to distribution differences: DistDiff
— Decreased Annotation requirements: AnReq
— Improved Relation Classification: RCls. This refers to models that are able to classify an object pair
— Improved Relevance Classification: ReleCls

Method	ClsNoise	Scale	Fit	Bias	AnReq	RCls	ReleCls
Data re-balancing	+		+	-			
Regularization	+		-				
Transfer Learning	+	+					
Knowledge distillation	+		+				
Rule distillation	+	+	+	+	+		
Data Augmentation		+	+	+	+		
Multimodal learning	+	+		-	+		
Self-supervised learning		+	+	+	+		
Knowledge graphs	+	+		-			
Metric learning		+					
Low-level dependencies	+		+			+	
High-level dependencies		+	+	-		+	
Semantic features	+	+		-	+	+	
Relevance estimation				-	-		+
Relevance prediction				-	-		+
Structural Ranking	+	+		-	-		+
Relation filtering				+			+

Table 2.8.1 – Visual Relation Detection methods

First and foremost, all studied VRD methods except [112] are fully-supervised methods where the model is trained with annotated images which provide for ground-truth relations with bounding boxes during the training phase. This conditions the

methods used but fully-supervised learning is a very efficient training method when sufficient data is available.

Second, as most methods, we focus on improving the estimation of the loss function, instead of improving its optimization. As we show in Chapter 3, due to the Human Reporting Bias, relation classes are heavily biased. Furthermore they have a high variance, thus many methods propose to use internal or external knowledge, which smooth estimated distributions consistently with the training data and external knowledge, and hence improve their estimation.

Relation Classification In order to improve Relation Classification, we aim to better fit the relation classifier. For this, external data is available in the form of word embeddings [97], synonym dictionaries and knowledge graphs. We show that **Rule Distillation** enforces rules on our model such that predicted relations are more consistent to the used external data which improves the estimation of the objective function. Furthermore, we explore **Metric Learning** in the context of relation classification, to make the representation of relations generalize better and decrease the number of examples required to learn one relation. We adapt the method introduced by Kaiser *et al.* [65] in order to take into account the polysemy and synonymy of relations.

Relevance Classification For Relevance Classification, the amount of available external data is very low as it requires additional relations on pair of objects in order to extract a probability distribution on the relevance of the pair. Furthermore, the relevance variable is hard to capture due to a high level of noise because of the Human Reporting Bias and the annotation process. Thus methods which are susceptible to underfitting are excluded. Finally, we lack external data on the ground-truth of relation relevance, which could be acquired by asking human annotators to annotate additional relations and rate relevance. This leads us to the conclusion that learning to detect relevant relations requires the integration of internal knowledge into the network. Among existing methods, only Relational Embedding [154] takes this fact into account when predicting relevance. Similarly to them, we explore **self-supervised learning** methods because they do not require additional data and have been proven useful in many different settings. We also explore high-level dependencies, in the form of statistics-based potentials, which helps capture relations between variables.

2.9 Conclusion and Contributions

In light of the issues related to the task of Visual Relation Detection and to the broader task of training deep neural networks on real world datasets, we argue that visual information is not sufficient to discriminate between relations. By considering the additional knowledge of relation similarities and focusing on relevant relations, this issue can be alleviated. Specifically, the major contributions of this work are as follow:

— **Human reporting bias in VRD datasets**: Similarly to the Wolf/Husky classification [121], we show that VRD datasets have exploitable biases that are not apparent due to the used evaluation metrics. These biases come mostly from a high imbalance in available annotated examples, and a high dependency between objects involved in the relation and the corresponding relation class. We show how this impact the detections of a competitive baseline and propose a metric as well as a new dataset to better evaluate the performance of existing methods.

— **Overcoming Relation Imbalance with Semantic Modelling**: This chapter focuses on the integration of knowledge intro a VRD model during training. This model is trained so that relations that are semantically, visually and spatially similar have similar probabilities. Two methods are proposed: the first method relies on a k Nearest Neighbor approach to train a deep neural network, in order to improve uncommon relation classification and take into account the structure of relations. For the second, external knowledge is used to improve the accuracy of the risk estimation during the training phase, aiming to increase the generalization capabilities of the model.

— **Relations Relevance**: In this chapter, we explore the inference phase and aim to tackle one aspect not directly tackled while training the model. A new scene graph construction method is introduced, integrating a learnt relevance criterion. In the absence of annotations for this criterion, two methods are proposed in order to focus on object pairs frequently related in similar contexts. The first relies on self-supervision and the second on integrating high-level dependencies between concepts into the model predictions. The impact of these methods is analyzed showing that the constructed scene graphs contain more uncommon relations while keeping a high overall recall. Furthermore, we find that this additional factor allows our model to predict relations on fewer and more relevant object pairs.

Chapter 3

Human Reporting Bias in Relation Annotations

In this Chapter, we first show that Visual Genome [71], the benchmark dataset on which most recent works are evaluated, exhibits several biases which makes training and evaluation of these works a difficult task. This will be a defining element throughout this work and will motivate our choices. Furthermore, we propose to use an additional evaluation metric highlighting the ability of the model to detect less frequent relations. Finally, we propose a new split of Visual Genome [71] in order to reduce the impact of this bias in the model evaluation.

3.1 Definition and Evidence in Visual Genome

The task of extracting a scene graph is a combinatorial problem, as the number of object pairs increases quadratically with respect to the number of objects and each pair can usually be described by several relations. Furthermore, images are very rich sources of information. Moreover, as pointed in [99], much of this information is not considered relevant by humans as it is redundant with prior knowledge. This makes people tend not to mention parts of them when describing images, omitting attributes that are "obvious or typical". For example, in Figure 3.1.1, several hundred relations are true, among which "*bed in front of wall*", "*bed near wall*", "*doctor has jacket*", "*doctor in shirt*", "*picture below picture*", "*pen hanging from jacket*". However in Visual Genome [71], the annotated relations are "*man ON bed*", *heart on picture*, "*picture on wall*", "*pants on man*", "*shirt on man*", "*diagram of heart*", "*doctor wearing lab jacket*", "*picture has heart*", "*man laying on table*", "*man covered with sheet*", "*doctor wearing shirt*", "*pictures on xray wall*", "*man pointing at skeleton*", "*man wearing shirt*" and "*light hanging over xray board*".

Figure 3.1.1 – Image example from Visual Genome. Several hundred relations are true but provide little understanding of the image.

In this section, we define the human reporting bias. We show that it impacts

the data distribution in Visual Genome. The resulting distribution prevents existing models from detecting several classes of relations. Furthermore, it prevents the evaluation of these models from reflecting how they generalize to unseen images. Then we propose a competitive baseline model and show the impact of this phenomenon on this model.

3.1.1 Definition

The reporting bias was introduced in [142]:

Definition 3.1.1 ***Reporting bias*** *is the phenomenon whereby the likelihood that situations of a certain type are described in text does not correspond to their relative likelihood in the world.*

In our case, due to reporting bias, relation concepts that are visible in images are not always mentioned by humans describing said images. The combinatorial nature of Visual Relation Detection is an amplifier of this phenomenon, as the very high number of possible annotations for a single image makes the frequency of true relations much higher than that of annotated relations. On the other hand, the difficulty to separate relations is amplified by this bias, as many true relations will not be annotated. Similarly, the under representation of many relation classes is a consequence of this bias, as humans tend to annotate a small set of concepts which creates the long tail distribution of both objects and relations examples per class.

In the case of relation annotations, this results in two types of imbalance:

1. Importance imbalance: Several relations are true for one object pair but are not annotated because they are not considered relevant in the current context.
2. Concept imbalance: Humans tend to use the same class of relations, even though many are true.

These two phenomena drive choices in this work. In this chapter, we show that they prevent models from learning representations that separate irrelevant relations and different relations between them. Furthermore, they prevent the assessment of existing datasets from reflecting how they would perform on new images.

3.1.2 Reporting Bias in Visual Genome

Visual Genome [71] is a recent dataset involved in the most recent works on VRD [58, 82, 154, 158, 165, 166, 170]. Here we show that annotations from Visual Genome are impacted by the reporting bias.

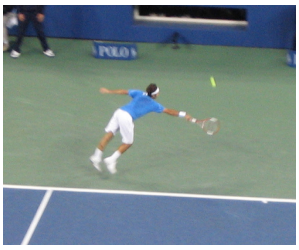
In Visual Genome, annotated relations are influenced by the annotation process. This process is defined as follows. First, annotators are asked to annotate regions of the image with a description in natural language, which we call Human Description Labels (HDL). Then one annotator extracts from each region the objects mentioned in it. Finally, annotators are presented with region descriptions and the corresponding objects and are asked what relations connect the objects, called Human Relation Labels (HRL), based on the region description. This introduces an important bias (which we quantify in the next Section) of which we show examples in Figure 3.1.2. Image **(a)** shows an example where a region was annotated on the pair *(woman, tennis)* and a relation was extracted from this pair. Image **(c)** shows a similar situation where the relation was not extracted from the region description. This suggests that the choice of relation to annotate is impacted not only by object characteristics but also by many factors which are hard to identify. The chosen regions, words and relations can be affected not only by the context of the object pair, their positions in the image, their sizes, but also the annotator vocabulary, knowledge, biases and so on.

(a) Woman is holding tennis racket.

	HDL	HRL	Visual
holding	✓	✓	✓

(b) Young man playing tennis.

	HDL	HRL	Visual
holding	✗	✗	✓



(c) Man holding a racket.

	HDL	HRL	Visual
holding	✓	✗	✓

(d) Hands holding tennis racket.

	HDL	HRL	Visual
holding	✓	✓	✓

Figure 3.1.2 – Human Description Labels (HDL) and Human Relation Labels (HRL) capture only some concepts visible in the image. Descriptions and relations do not always coincide. In all images, the concept *holding* is present in the image. In **(a)**, **(b)** and **(d)** *holding* is mentioned in description. In **(a)** and **(d)**, *holding* is mentioned in relation labels, however, in **(d)**, the extracted relation is (*hands, holding, tennis racket*) instead of (*man, holding, tennis racket*). In **(c)**, *holding* is mentioned in a region description but not in a relation label. In **(b)**, a similar situation is described as *young man playing tennis* and the concept *holding* is not mentioned.

3.2 VRD Models Evaluation

3.2.1 Relation Imbalance

To show the impact of the reporting bias, we show how it impacts the relation distribution and creates a relation imbalance. Results on existing methods [23, 58, 80, 89, 102, 158, 164, 166, 170] trained on Visual Genome [71] have shown that most predictions of the model are *on* and *has* relations. One reason for this comes from the distribution shown in Figure 3.2.1, considering examples for the 20 most frequent relations in VG. We study the most used split of Visual Genome, introduced in [158], called VG-IMP.

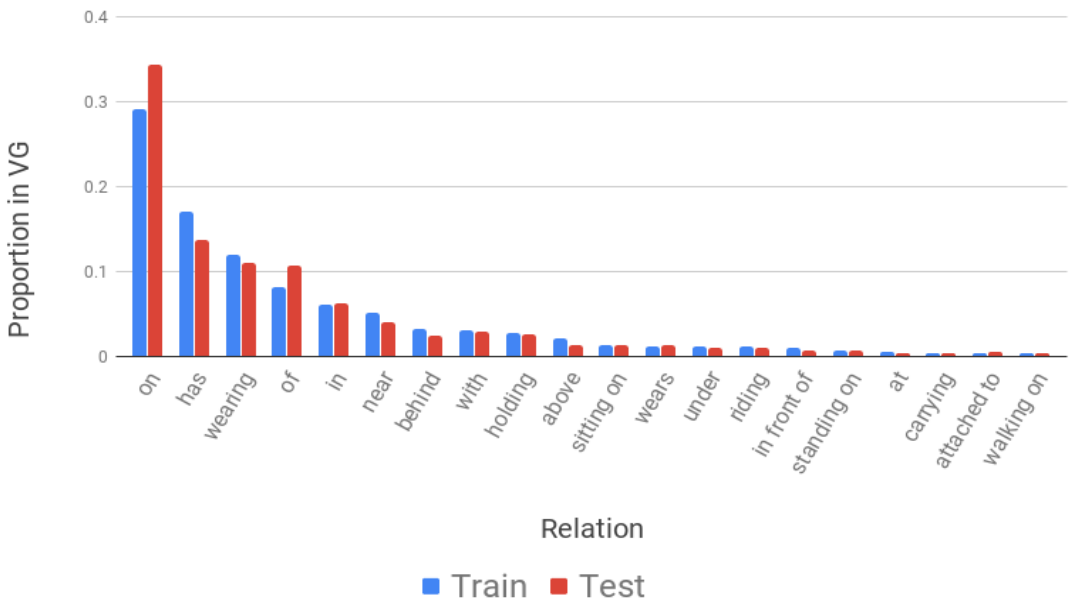


Figure 3.2.1 – Proportion of examples in VG with the VG-IMP split [158] sorted by proportion in Train.

First, for both train and test sets, the relation *on* is the most frequent relation, with around 30% of all annotated relations. Furthermore, the number of examples decreases rapidly, and only 14 classes are represented by more than 1% of the dataset. This raises an important consideration, related to the observation that for many tasks, neural networks are able to learn correlations between features but not causal relations. In this case, training VRD models on these skewed datasets results in biased models which will tend to predict the most common relation. On the one hand, this is not necessarily a drawback, as these interactions tend to occur frequently in general cases and these biases can be interpreted as a form of common sense knowledge, e.g. for the pair (*person*, *car*) relations *in* and *drive* are most likely. However, on the other hand, these biases may prevent learning the representation

of less frequent relations and what makes them distinct. Furthermore, the 4 most frequent relations represent together two thirds of annotations and the 11 most frequent ones represent together more than 90% of them. This makes the retrieval of the other relations have a very low impact on the evaluation of models.

This bias is more visible at the level of object pairs. To study this, we extract a set of object categories associated with each object class from WordNet [34] hypernyms. The proportion of relations is measured for each pair of categories in Figure 3.2.2. It shows that for most pairs, the most frequent relation holds a large majority of examples.

Distributions differ for a given pair of categories between train and test splits. The category with the most differing distributions is *vehicle* - *artifact* for which the relation *on* is the majority in the train set while it is the relation *has* in the test set. However, for all other object pairs, the majority class remains the same between splits. This makes predicting the most frequent relation for the given pair an efficient strategy, as pointed in [166], where authors show that predicting the most frequent relations of the object pair under study is enough to predict the correct relation 75% of the time. It is correct 95% of the time when predicting the 5 most frequent relations. For images with few objects, this alone will provide a good recall at 100 predictions.

Thus, in order to keep a high precision, the main difficulty of this task becomes the reliable detecting of object classes and selection of which object pairs to annotate, *i.e.* the selection of object pairs relevant to a human observer. In this work, we mainly consider the task of relation classification, assuming perfect object detectors. Thus, we focus on the tasks of detecting and classifying relevant relations with a high precision, especially of rarer relation classes.

Finally, this also shows another aspect of the reporting bias. Indeed, these category pairs have the advantage of removing relation polysemy: for a given object pair, one relation class always keeps the same meaning. Thus, for the pair *person* - *clothing*, the four most common relations *wearing, has, in* and *wears* have the same meaning even though they belong to different classes. When modelling these separately, this introduces a high amount of noise into the network, making learning to separate these relations difficult.

3.2.2 Evaluating Relation Diversity

To take into account the specificity of the problem, such as the inherent incompleteness of available annotation, we measure the *recall@k* as in [89]. Thus we measure the fraction of ground truth annotations among the top k relations retrieved

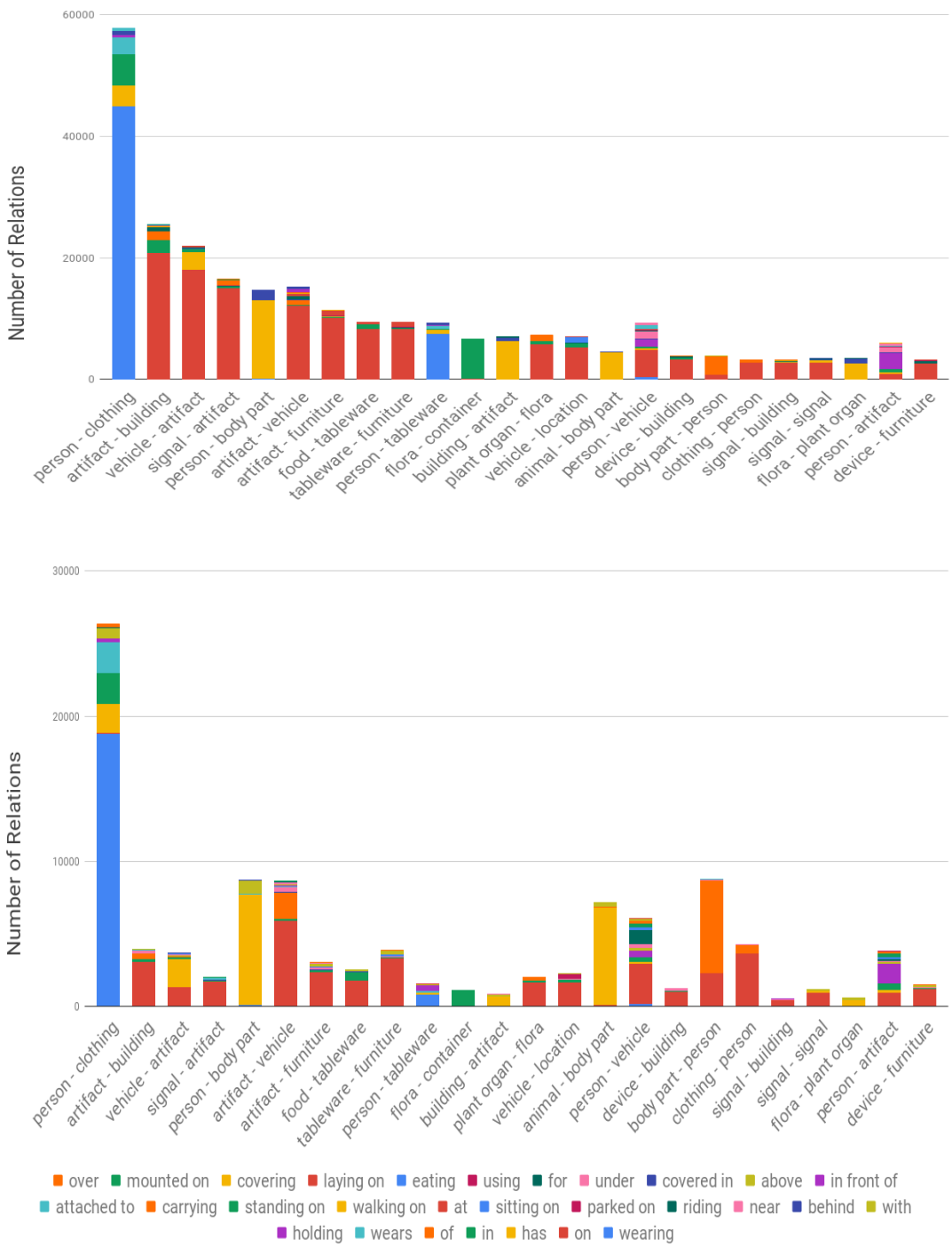


Figure 3.2.2 – Relation distribution by pair of object categories in both train and test sets of VG-IMP [158]. For most pairs, the most frequent relation holds a large majority of examples. Furthermore, the most frequent relations remains the same between train and test sets in all cases but *(vehicle-artifact)*.

by the model. Most recent works [23, 58, 80, 89, 102, 158, 164, 166, 170] focus on the model performance on the image macro recall, which takes the average recall

over each image:

$$\text{ImMacro } R@k = \frac{1}{|D|} \sum_{I \in D} R@k_I \tag{3.1}$$

where $R@k_I$ is the recall for image I.

However, the ability of the model to generalize also depends on the diversity of retrieved relations, which is not captured by the previous metrics due to dataset imbalance. Indeed, the relation 'on' represents 30% of the annotated relational phrases in Visual Genome. Furthermore, retrieving uncommon relations is necessary for many applications where important classes have a low number of training examples. To complement this evaluation, we propose to also use class macro recall, which averages the recall over each relation class:

$$\text{ClsMacro } R@k = \frac{1}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} R@k_r \tag{3.2}$$

where $R@k_r$ is the recall of relations of class r. The underlying intuition here, is that by averaging on every example, the image macro recall exhibits the ability of the network to reliably retrieve most relations. Our focus here is for our network to both reliably retrieve most relations but also focus on relations that occur more infrequently and differentiate images from another.

3.3 Visual Relation Detection

In this section, we present a model able to detect visual relations using an annotated dataset and show how the previously described relation imbalance impacts its outputs and performance.

3.3.1 Object and relation classification

We follow the same steps as in Section 2.6.1. A Convolutional Neural Network (CNN) is used to extract image representation and object region proposals as displayed in Figure 3.3.1. For each pair of objects, region representations are extracted from the image feature map using ROI-Pooling [120] (a). Following [166], we add the visual representation of the union bounding box to spatial features extracted from the binary masks using a two-later CNN (b). Then the representations of head, tail and union bounding box are passed through two feed-forward layers. They are noted $\boldsymbol{f}_h$, $\boldsymbol{f}_t$ and $\boldsymbol{f}_{h \to t}$.

For object classification, the parametric probability distribution function p_θ^o is defined as

$$p_\theta^o : \boldsymbol{f} \to \text{softmax}(\boldsymbol{W}_o^{vis} \boldsymbol{f}) \tag{3.3}$$

Similarly to [41, 170], for relation classification, four different streams contribute to each class score. Indeed, [41] noted that the head and tail streams help focus on surrounding objects and human body parts to disambiguate relations. Thus, the head, tail, spatial and visual streams are trained to predict relation scores (logits). These scores are then summed and passed through a softmax layer. The relation probability distribution function is defined p^r_θ:

$$\begin{aligned} \boldsymbol{y}^{vis}_{h\to t} &= \boldsymbol{W}^{vis}_t \boldsymbol{f}_t + \boldsymbol{W}^{vis}_h \boldsymbol{f}_h + \boldsymbol{W}^{vis}_e [\boldsymbol{f}_h, \boldsymbol{f}^{vis}_{h\to t}, \boldsymbol{f}_t] \\ \boldsymbol{y}^{spat}_{h\to t} &= \boldsymbol{W}^{spat}_r \boldsymbol{f}^{spat}_{h\to t} \\ p^r_\theta : (\boldsymbol{y}^{vis}_{h\to t}, \boldsymbol{y}^{spat}_{h\to t}, v_h, v_t) &\to \text{softmax}\big(\boldsymbol{y}^{vis}_{h\to t} + \boldsymbol{y}^{spat}_{h\to t} + \log \psi(v_h, v_t)\big) \end{aligned} \tag{3.4}$$

where $\boldsymbol{W}^{vis}$ and $\boldsymbol{W}^{spat}$ weights project the head, edge and tails of the relation into the relation space. All vectors are then summed and the softmax function is applied to the sum to compute the probability over the set of relations, including the *null* relation $\varnothing$. Similarly to the semantic bias used in [166], $\psi(v_i, v_j)$ is a prior potential computed by measuring the frequency of each relation for each pair of object classes in the training dataset:

$$\psi(v_h, v_t) = \left(\frac{\sum_I \sum_{i\neq j} \mathbb{1}[v_i = v_h, v_j = v_t, e_{i\to j} = r]}{\sum_I \sum_{i\neq j} \mathbb{1}[v_i = v_h, v_j = v_t]} \right)_{r\in\mathcal{R}} \tag{3.5}$$

Thus, in equations 3.3 and 3.4, object and relation classes are assumed exclusive.

These probability distributions are used as object and relation classifiers. In order to translate these probabilities into a Scene Graph, the most probable labels for nodes and edges are selected. However, with one selected label per node and edge, the number of annotated relations in image I would be equal to $n(n-1)$. On Visual Genome, the average number of objects is 21.2. An image with 21 objects has 420 object pairs. Thus annotating the most probable relation for all these pairs would not be useful to a human and create many false negatives, as the average number of relations per images is 17.7. Therefore, the choice for most recent works [23, 58, 154, 158, 166] is to select the top k scoring relations. This score is computed as

$$p^o_\theta(v_{h,k}|\boldsymbol{f}_h) * p(e_{h,t,m}|\boldsymbol{f}_{h,t}) * p(v_{t,l}|\boldsymbol{f}_t) \tag{3.6}$$

for $k, l \in [1 \dots n_\mathcal{C}]$, $m \in [1 \dots n_\mathcal{R}]$, $h, t \in [1 \dots n]$, $h \neq t$. This ranking process has drawbacks, such as outputting scene graphs with a fixed-size for every image, contrary to a threshold-based method. We use the same method to be able to compare with existing methods and leave the study of threshold-based methods for further research. These methods could increase the precision of the studied models by taking into account the amount of information visible in each image.

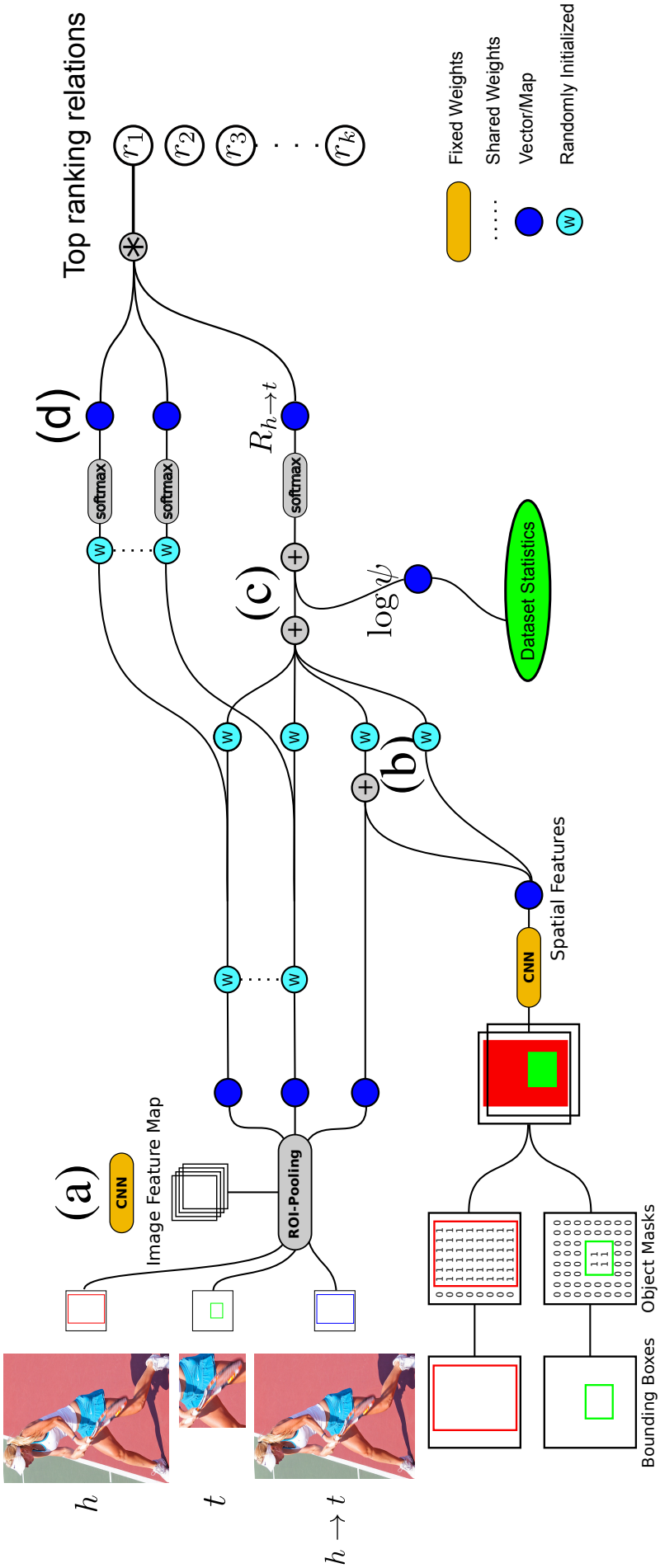


Figure 3.3.1 – Visual Relation Detection Baseline. For each pair of objects, visual and spatial representations are extracted by convolutional layers ((**a**) and (**b**)). From each representation, scores (logits) are computed and summed. They are added to a statistics-based prior potential (**c**) and passed through a softmax layer in order to obtain a distribution probability over the set of relations. In parallel, probability distributions are computed for each object (**d**).

3.3.2 Training

Objective function The model parameters are defined as

$$\theta = \arg\max_{\theta} \mathcal{L}(\theta, \mathcal{D}) \tag{3.7}$$

$$= \arg\max_{\theta} \alpha\mathcal{L}_o(\theta, \mathcal{D}) + \beta\mathcal{L}_r(\theta, \mathcal{D}) \tag{3.8}$$

where α and β are weights, set to 1 following [166]. The object classification loss $\mathcal{L}_o(\theta, \mathcal{D})$ is defined as

$$\mathcal{L}_o(\theta, \mathcal{D}) = \sum_{(I,y)\in\mathcal{D}} \sum_{i=1}^{n_I} \sum_{k=1}^{n_\mathcal{C}} \delta_k(y_i) \log p_\theta^o(v_{i,k}|x) \tag{3.9}$$

where $p_\theta^o(v_{i,k}|x)$ is computed using branches in **(e)**, $\delta_k(y_i) = 1$ if $y_i = k$ and 0 otherwise and n_I is the number of sample object annotations.

The relation classification loss $\mathcal{L}_r(\theta, \mathcal{D})$ is defined similarly as

$$\mathcal{L}_r(\theta, \mathcal{D}) = \sum_{(I,y)\in\mathcal{D}} \sum_{h=1}^{n_I} \sum_{t=1, t\neq h}^{n_I} \sum_{k=1}^{n_\mathcal{R}} \delta_k(y_{h,t}) \log p_\theta^r(v_{h,t,k}|x) \tag{3.10}$$

Implementation details In the sequel, we present in details how several hyper-parameters are chosen.

- Object sample selection: During training, 256 regions of interest (ROI) are selected. Among them
  - 25% are positive annotation: either
    - a ground truth object annotation
    - a region with IOU (intersection over union) with a ground truth object within two thresholds
  - 75% are negative annotations, with IOU outside those thresholds
- Relation sample selection: Similarly, at most 256 object pairs are selected, with 25% positive relations sampled from object pairs with at least one annotated relation and 75% negative relations sampled from object pairs with no annotated relations.
- For this work, we focus on the relation classification branch thus we use a standard object detection and classification framework. To provide fair comparisons to existing works, the Faster R-CNN [120] framework is used with a VGG network [131] for object detection and the extraction of region representations by ROI-pooling. It is pre-trained on MS-COCO [86] and during training for VRD, all convolutional layers are frozen.
- Training is done by Stochastic Gradient Descent, as described in Section 2.1

with Momentum [115], with momentum factor equal to 0.9.

3.3.3 Model evolution during Training

In this Section, we study the optimization of the training loss on VG-IMP [158] and how the evolution of the loss translates into the target metric. To prevent overfitting, training is stopped when a criterion has stopped increasing. The criterion used in existing works is the SGCls (as defined in Section) recall at 100 propositions, $R@100$. It is evaluated on a validation set of 5,000 images. In Figure 3.3.2, the evolution of the cross-entropy loss over 100 iterations is shown in blue, red dots are averages over each epoch and yellow dots the standard deviation over the last epoch. After 2 epochs, the average training loss remains constant, and its standard deviation does not decrease after four epochs.

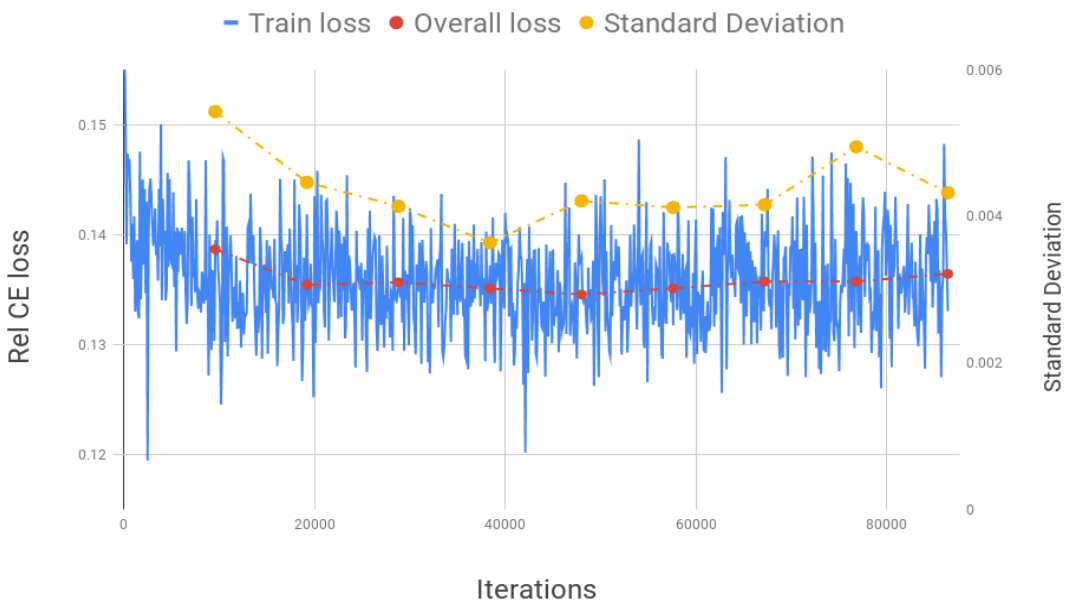


Figure 3.3.2 – Cross-entropy loss for the baseline over 9 epochs. Blue points are the average over the last 100 iterations and red dots are the average loss over the last epoch (9600 iterations).

The constant standard deviation and mean suggest that the network is not able to fully capture the variance of the data.

Results Variance To show the limits of this training procedure, performance of the model on all three splits (*i.e.* train, validation and test) are measured. Since the number and distribution of samples vary between each dataset, we also study whether those results are statistically significant. However, iterating over all mini-batches of VG-IMP [158] once takes around 60 minutes. Thus, testing the variance

of results by training several models with different parameter initialization would be very time consuming. Thus, it is instead tested by bootstrapping on the set under study:

1. Sample 80% of the set by randomly choosing an image with replacement.
2. Compute the Recall over all selected images.

Each step is done 1,000 times. The motivation behind this is to estimate the variability of results for a single model on sets with similar distributions to the test set. The limits to this method is that different training processes of the model would increase this variability.

Figure 3.3.3 shows the performance of the model after each epoch on the whole train dataset. Contrary to existing works which evaluate only the SGCls performance to select when to stop the training, we use RelCls instead. These results contrast with the cross-entropy loss shown in 3.3.2. They show that the network is unable to capture the variance of relations, but also that the performance on relation classification deteriorates after epoch 4. This shows that the SGCls performance does not reflect the ability of the model to generalize on the task of RelCls and that using it as a stopping criterion results in overfitting. Thus we use RelCls.

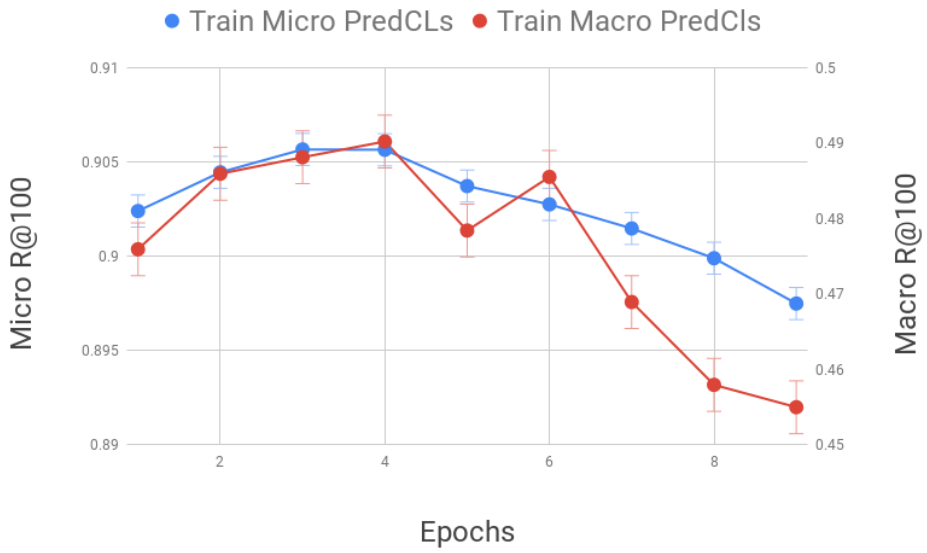


Figure 3.3.3 – RelCls performance of the baseline on the train set of VG-IMP [158] at the end of each epoch. Both performance metrics deteriorate after 4 epochs. This shows that the optimization of the objective function does not result in increased performance on the target task.

This highlights that the cross-entropy loss does not correlate with the target task performance. Results on the validation and test splits are displayed in Figures 3.3.4 and 3.3.5. They show similar trends with decreasing performance. Indeed, in Table

3.3.1 results for different numbers of epochs are all but the first contained in the 95% confidence interval.

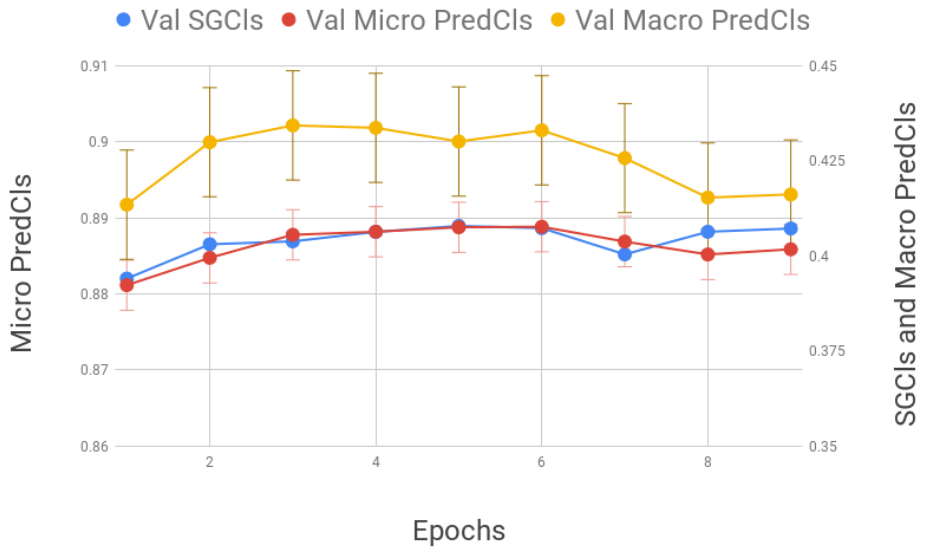


Figure 3.3.4 – SGCls (blue and right), Micro (red and left) RelCls performance of the baseline on the **Validation** set at the end of each epoch. Performance on the two tasks are not directly correlated.

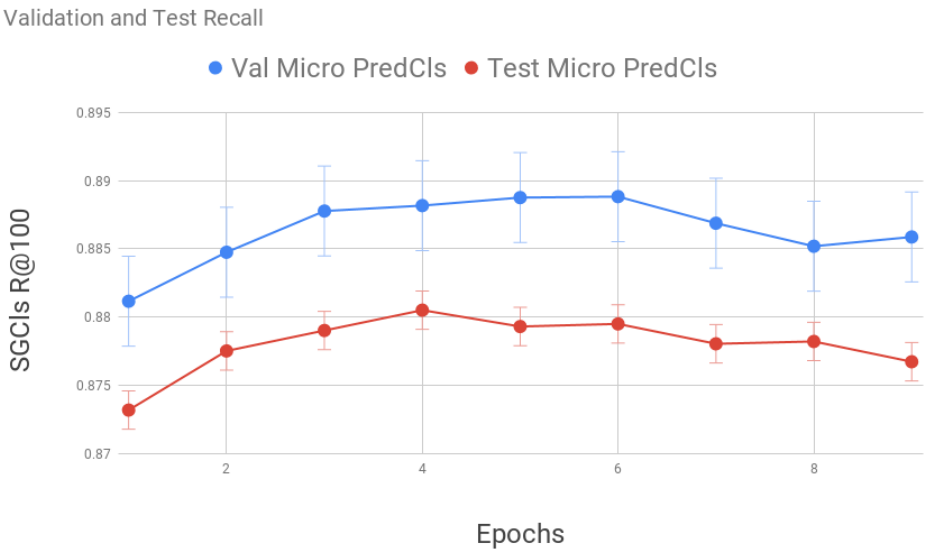


Figure 3.3.5 – **Validation** and **Test** performance of the baseline at the end of each epoch. The performance on each split suggest that they follow similar distributions as they follow similar trends after 6 epochs where each slightly decreases.

Micro R@k	2.5%	Median	97.5%
50	80.4	80.8	81.1
100	87.8	88.05	88.3

Table 3.3.1 – Median, 97.5 and 2.5 centiles for RelCls R@k on the train dataset of the baseline after 3 epochs.

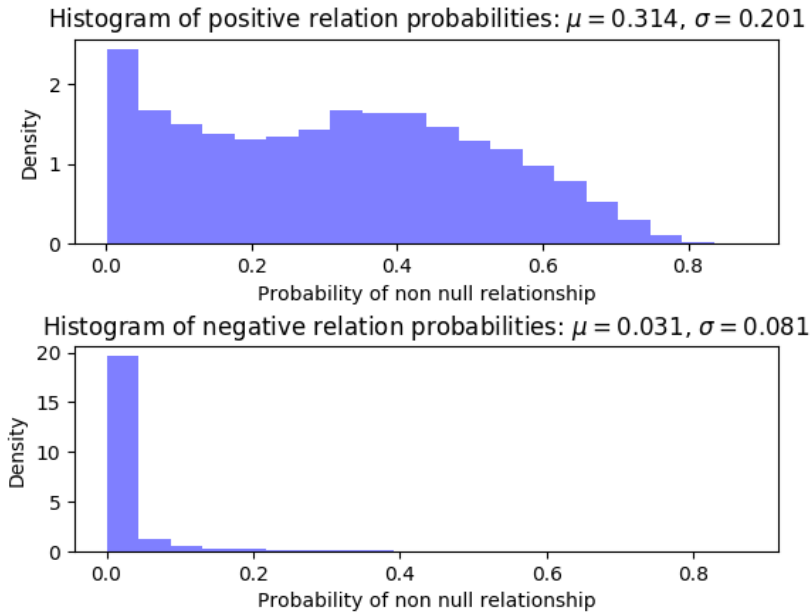


Figure 3.3.6 – Histogram of probability that any relation is true: $1 - p_\theta(R = \varnothing)$ for object pairs with at least one annotated relation (top) and no annotated relation (bottom).

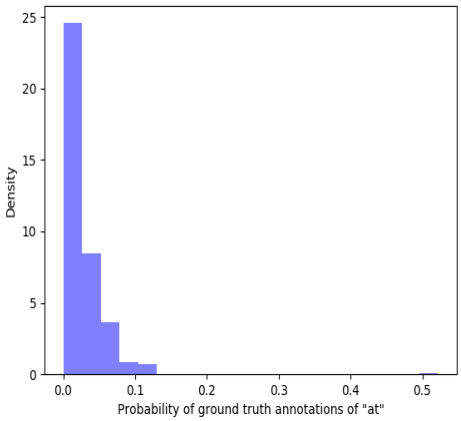
Relation Detection Figure 3.3.6 shows the probability that any relation is annotated given an object pair. The top graph shows this distribution for pairs where at least one relation has been annotated, which we call relevant, and the bottom one where no relation has been annotated. We notice that the majority of relevant relations have a probability lower than 0.5 of being annotated. Filtering out relation where $p(R = \varnothing) < 0.05$ would remove most of the irrelevant relation but also remove a high number of relevant ones, resulting in high precision but low recall outputs. Moreover, this shows that the network does not learn to reliably predict whether an object pair holds a relevant relation.

In Figure 3.3.7, output probabilities for the annotated relations of six classes are displayed. For probabilities of all classes, please refer to Appendix A. For most relations, the maximum probability density is reached between 0 and 0.1, except for relation *wearing*. Rarer classes such as *at, mounted on* and *eating* are rarely detected by the network. More frequent classes such as *has, on* and *wearing* have a significantly higher mean and by ranking relations by their respective probabilities

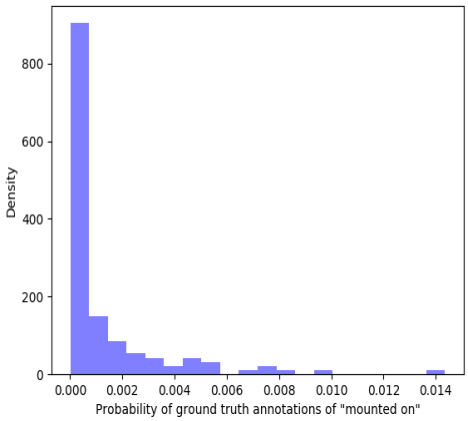
makes the model classify into those much more frequently than uncommon ones. For smaller scene graphs, these rarer relations have a very low probability of being predicted.

Finally, Figure 3.3.8 shows the confusion matrix of our Baseline. The top matrix represents the normalized confusion matrix. We notice that the model fails to detect most relations and classifies them as background (*i.e.* $\varnothing$ relation), except relations *on* and *wears*. This is a second piece of evidence which suggests that the model is unable to detect relevant relations.

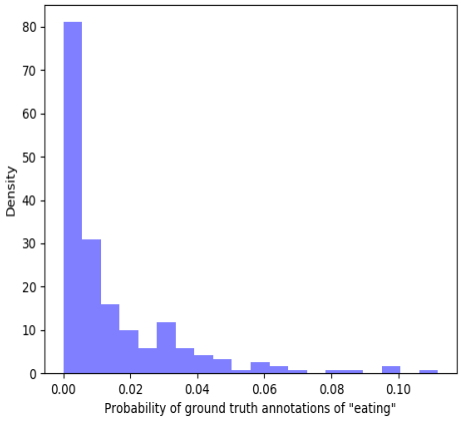
On the bottom figure, we display the confusion matrix without considering background class and instances. Instances of most relations are mis-classified as *on*, *has* or *wears*. This does not completely capture how the model detects relations because the test setting ranks classification scores instead of taking the maximum score for each pair of object. However, this suggests that the reporting bias has a negative impact on relation detection, as the most detected relations are the most frequent relations in Visual Genome.



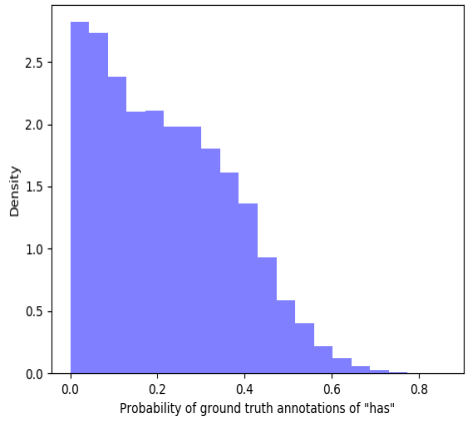
at



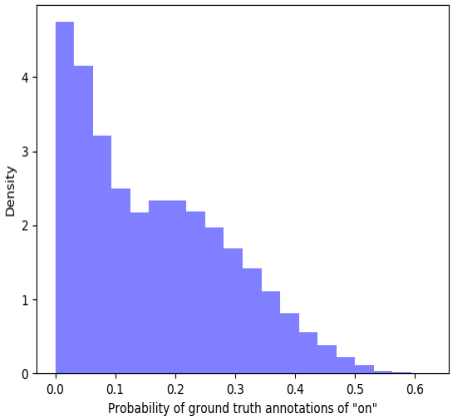
mounted on



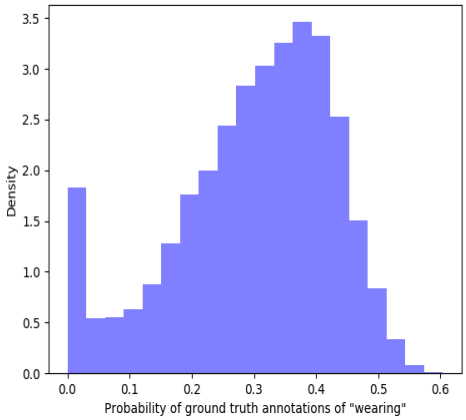
eating



has



on



wearing

Figure 3.3.7 – Distribution of baseline output probabilities for object pairs annotated with relations.

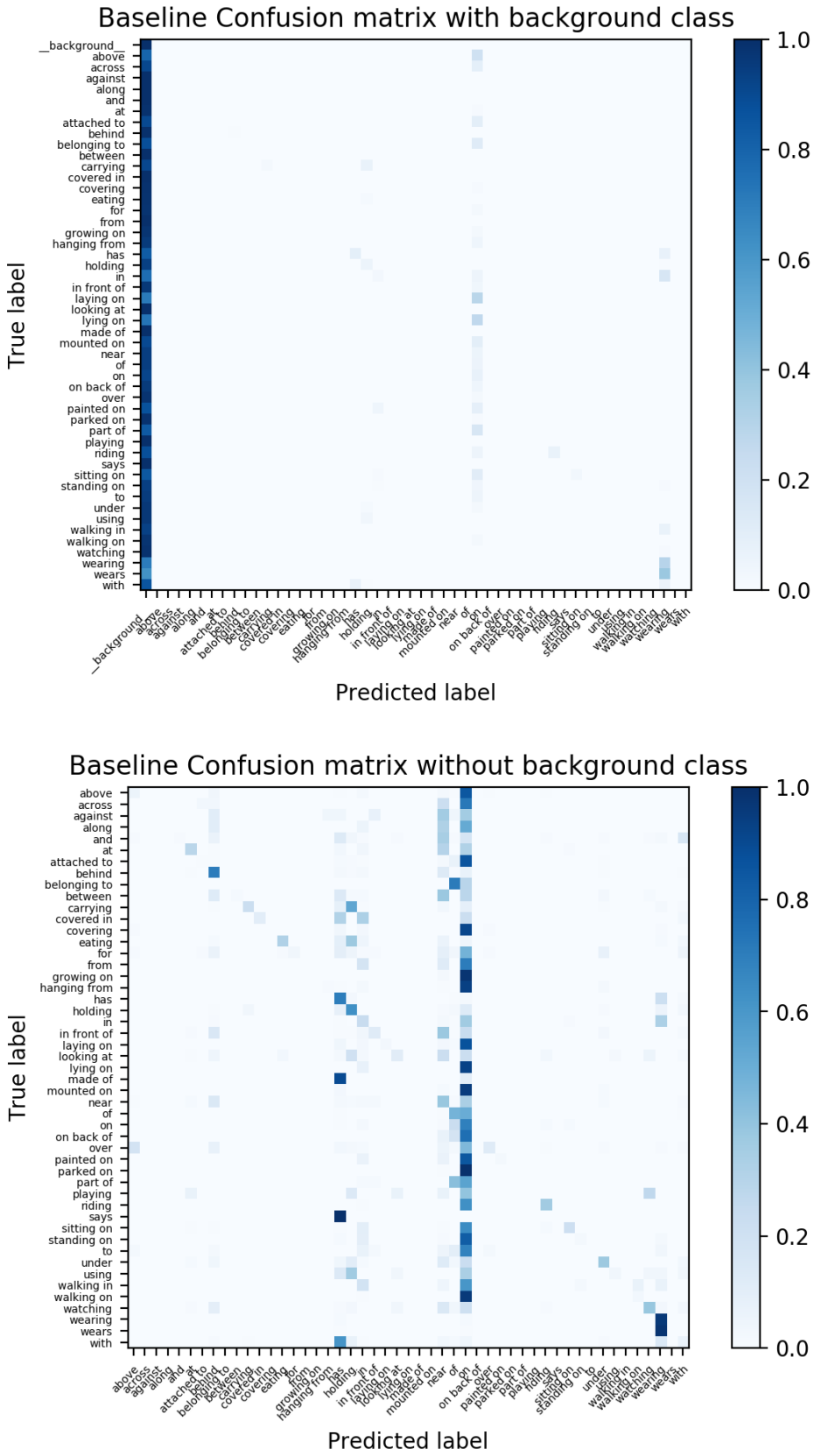


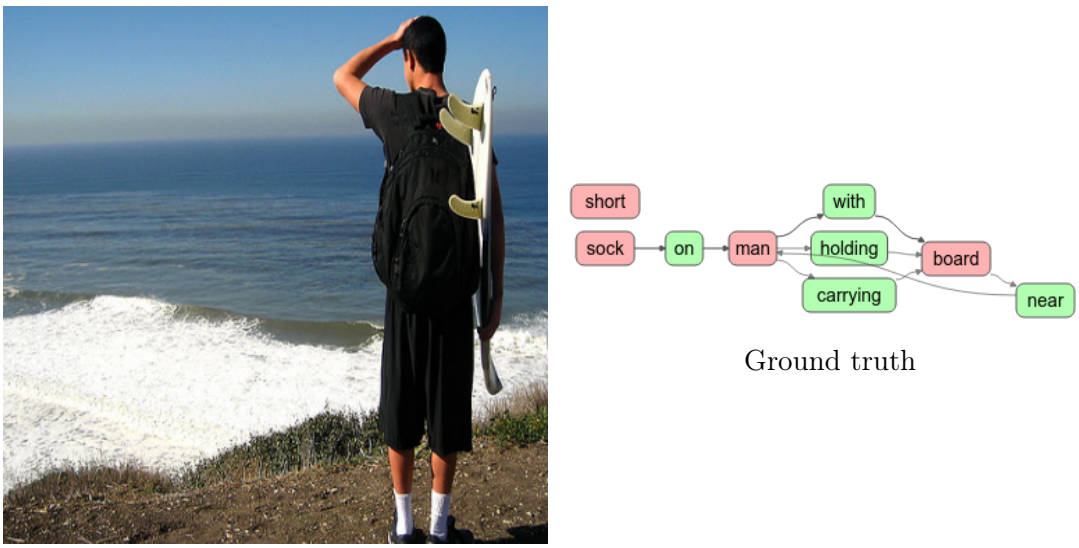
Figure 3.3.8 – Confusion Matrix of our Baseline. The matrix on *top* shows that most object pairs are classified as background and suggests that the model is unable to detect relevant relations. The matrix at the *bottom* suggests that the reporting bias has a negative impact on relation detection, as instances of most relations are mis-classified as "*on*", "*has*" or "*wears*".

Generated Scene Graphs Figures 3.3.9 and 3.3.10 show examples of images from Visual Genome. On Figure 3.3.9, due to the low number of objects, and to the correct estimation that the pair (*man, board*) is the more likely to hold a true relation, the network is able to predict all groud truth relations for this pair: *holding, with, carrying*. However, this also allows the prediction of false positives such as *riding* and *on*.

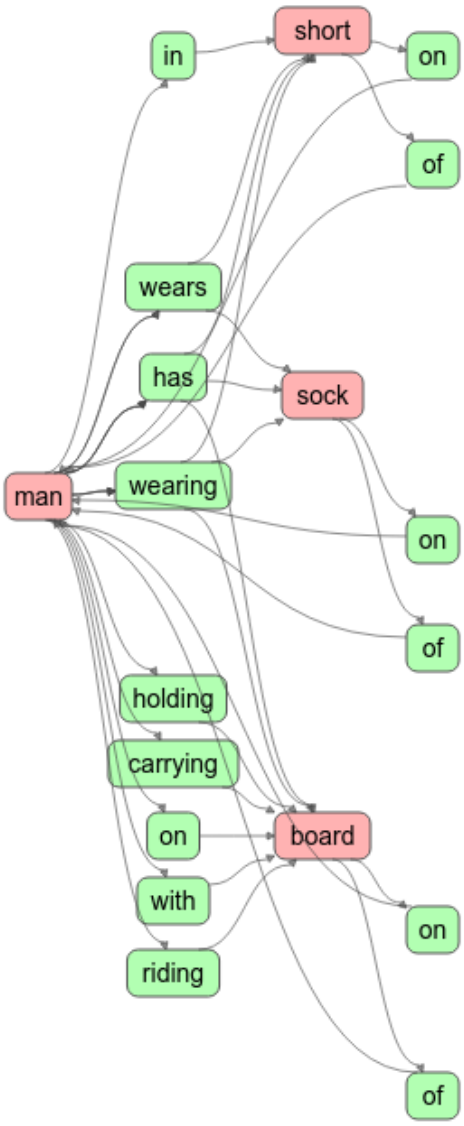
However, for images with many objects such as Figure 3.3.10, all predicted edges are labelled with relations *on* or *wearing*. Furthermore pairs with annotated relations, such as *(hand, man)* (*top*) and (*fence, street*) (*bottom*) have no predicted relation.

Conclusion While this method does not provide a definitive proof that the network does not learn beyond the first epoch, the decrease in performance on the training test after the fourth epoch is statistically significant. The stable loss suggests that the data variance is too high and that the network is underfitting; however the drop in RelCls performance would suggest that it is instead overfitting. These two hypotheses can be compatible, with the network overfitting on a region of the input space and underfitting on a different region of this space, where the model is not able to capture the variance of data.

1. The model performance on test deteriorates after several epochs.
2. Validation SGCls and Test RelCls are not correlated.
3. Model performance on the target measure, where the scene graph is constructed by selecting the top k relations does not correlate with the cross-entropy loss.
4. The model does not learn to detect uncommon relations and highly favors more common ones.

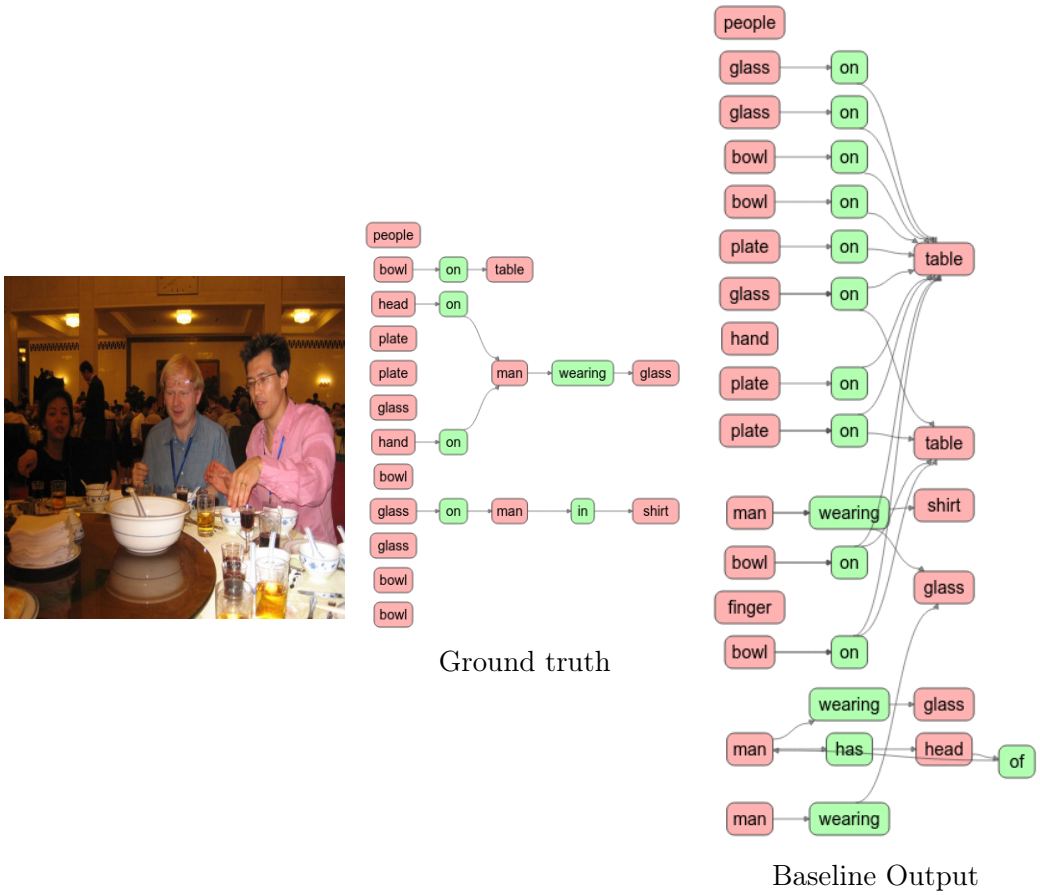


Ground truth



Baseline Output

Figure 3.3.9 – Ground truth annotations and outputs of our baseline on one images from Visual Genomes.



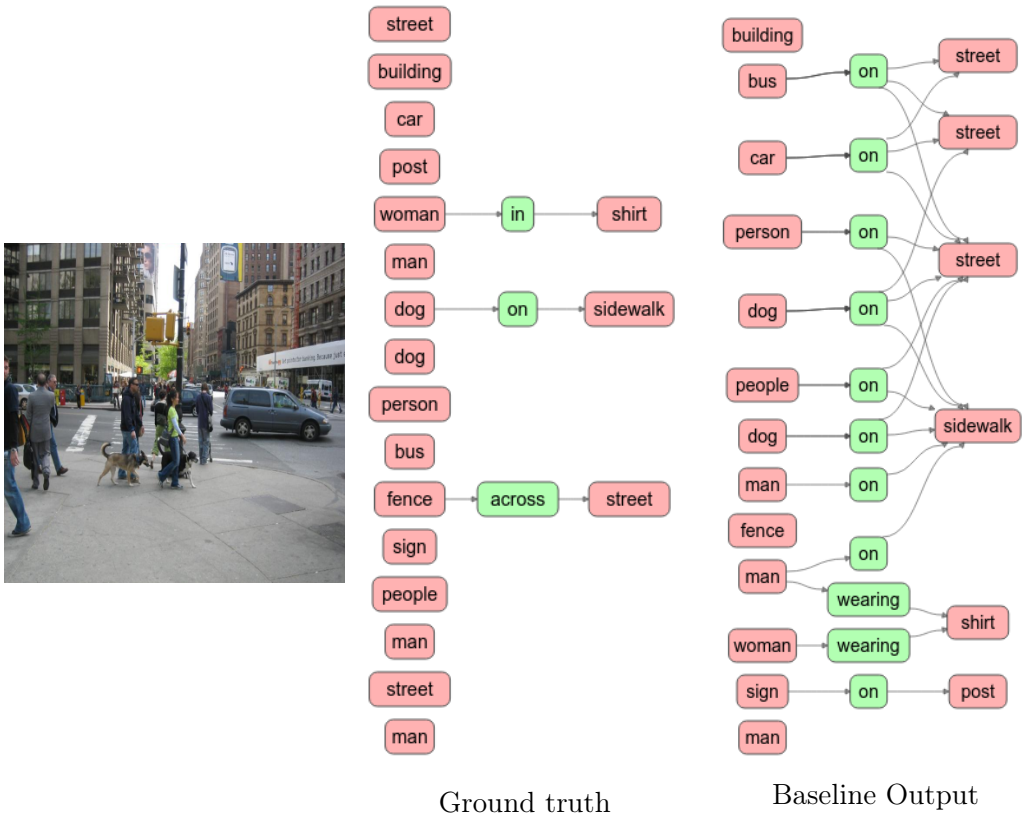


Figure 3.3.10 – Ground truth annotations and outputs of our baseline on two images from Visual Genomes.

3.3.4 Comparative study

In Table 3.3.2, we compare our proposed method to state of the art approaches. PIXEL2GRAPH [102] iteratively refines object and relationship heatmaps with a stacked hourglass network making use of global context. MESSAGE PASSING [158] iteratively learns to refine relationship and object representation by passing mesages through the scene graph. MOTIFNET [166] captures higher order correlations between objects and relationships using LSTM layers. SGP [58] is a permutation invariant graph predictor that refines predictions from MOTIFNET using attention over linguistic and visual neighbor features. The notable differences between our baseline and MOTIFNET are the absence of LSTM layers, the added addition of logits from four branches (Figure 3.3.1 **(c)**).

	SGCLS	RELCLS
	Micro	Micro
	R@100	R@100
PIXEL2GRAPH [102]	38.4	86.4
MESSAGE PASSING [158]	47.2	83.6
SGP [58]	**50.8**	88.2
MOTIFNET - NO CONTEXT [166]	46.3	86
MOTIFNET [166]	47.7	**88.3**
Baseline (Ours)	47.2 [46.7 - 47.7]	88.7 [88.4 - 88.9]

Table 3.3.2 – Results on VG. Recalls are in % and evaluated without scene graph constraints. Numbers between brackets correspond to the 95% confidence interval.

Table 3.3.2 shows that our baseline is competitive with the state of the art method and outperforms MOTIFNET - NO CONTEXT [166] by 2.7 points even though it has a lower number of parameters. We attribute this first to the removal of the outer product described in [166] between object semantic representations and relation visual representation, which provides a 2 points increase. Second, the added predictions for relations based on head, tail and spatial features are responsible for a 0.7 point increase. The performance of simple model is indeed deteriorated by this product, which we hypothesize results in too much information loss in the visual representation. For this reason, we keep only the visual and spatial representation of the union region, as shown in Figure 3.3.1. However, MOTIFNET performs better on SGCLS. This suggests that passing messages between object and relation classes is necessary to take into account the relations between constituents of the relations and their respective uncertainties.

3.4 Making the R in VRD matter

3.4.1 Motivation

In the same vein as Goyal et al. [55] who observe that the output of Visual Question Answering models is much more dependent on text priors than visual cues, we noted for existing models, that relation predictions are conditioned more by the object categories than by their relations in a specific image. This is not directly apparent in results as text distributions in Visual Genome [71] are heavily skewed. Figures 3.4.3a and 3.4.3b display the distribution of relations on Visual Genome for the most frequent pairs of object categories. They are computed by grouping object classes by manually defined hypernyms. They show that the majority relation for each pair often represents from 50% to 75% of the examples. This makes the predictions of models biased towards these relations and the evaluation does not reflect that. To make the image macro recall more sensitive to how the model generalizes on the task of relation classification, we propose a new split of Visual Genome [71] called VG-RMatters.

This contribution follows the same motivation as many recent works in Natural Language Processing that point out limitations in existing methods that are not directly apparent. They show that focusing on standard splits may result in Type-I errors [54], that annotating processes may introduce annotator bias and result in overestimation of model performance on several datasets [42] and that models tend to make use of language biases, ignoring visual information [55] or adopt heuristics valid in some case but not all [8, 96].

3.4.2 Dataset Definition

This split is defined such that predicting the most common relation of each pair of object categories is not a viable strategy. Thus the dataset should be such that the majority relation of each pair of object categories is different between the train dataset and the test dataset. The main difficulty towards this goal stems from the fact that mini-batches and thus splits are defined by grouping sets of images together instead of processing arbitrary sets of relations. Thus if an image contains one uncommon relation and a set of common relations, it will offset the distribution of all object pairs in the image, not only the one with the uncommon relation, as shown in Figure 3.4.1. This difficulty makes the distribution of relations in both sets hard to balance and, for a given dataset, this bounds the proportions of examples of each class.

Figure 3.4.1 – Example image with uncommon relation (*person, riding bicycle*). Adding this image to the test dataset also adds *(bicycle, on, pavement)*, *(green leaves, on, tree)*, *(bike rider, wearing, a white shirt)*, *(bike rider, wearing, shirt)*, *(tree, on side of, road)*, *(bush, on, street)*, increasing the number of *wearing* and *on* relations.

3.4.3 Dataset Statistics

In order to quantify the differences between datasets, we measure two quantities: the average proportion of the majority relation class and the average entropy for each pair of object categories. Table 3.4.1 shows the differences of those quantities between VG-IMP and VG-RMatters splits. For a uniform distribution, the entropy is 1.71 and majority proportion is 0.02. This shows that the distribution is very far from a uniform distribution but due to data limitations and object affordances (what actions can be performed on/with objects), it is not a realistic goal. Both metrics show that the diversity of relations for the VG-RMatters split is higher.

Split	Majority Proportion	Average Entropy
VG-IMP [158]	0.62	0.55
VG-RMatters	0.44	0.68

Table 3.4.1 – Entropy and proportion of the majority relation in VG-IMP [158] and VG-RMatters for the top 50 pairs of object categories (81% of examples). VG-RMatters shows a greater diversity of relations for the most frequent object pairs.

In Figure 3.4.3, the distributions of both splits are compared for the top 50 object categories and show that even though the most common relation does not necessarily change between both, relations are more varied for most categories. The

differences in distribution shown in Figure 3.4.3 and Table 3.4.1 suggest that this split tests the model's ability to generalize and will provide a better understanding of the model performance.

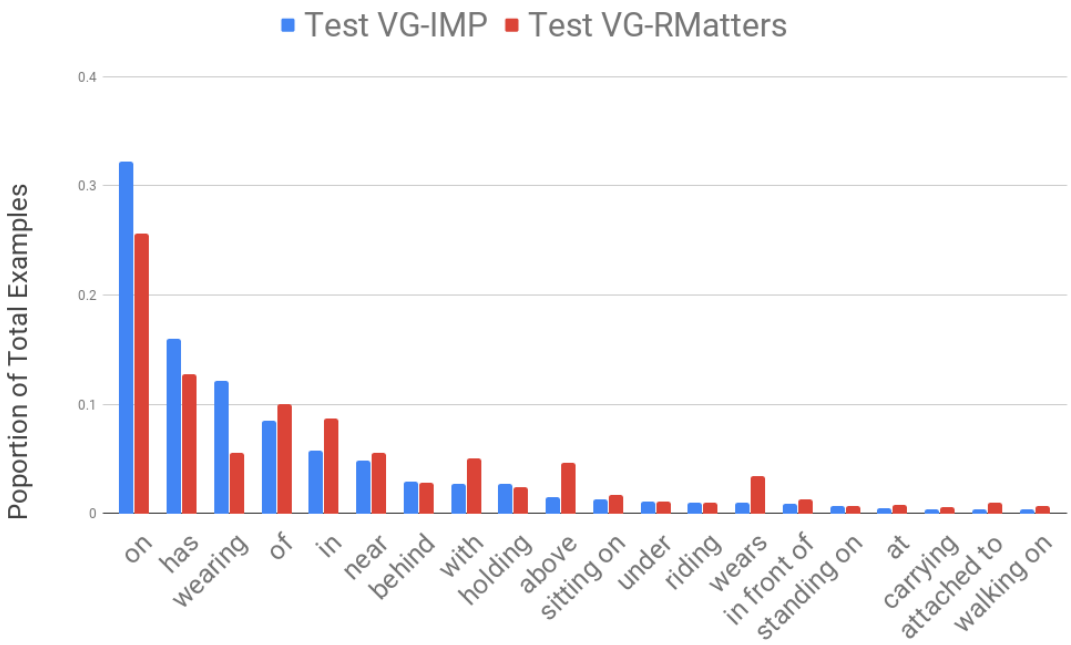
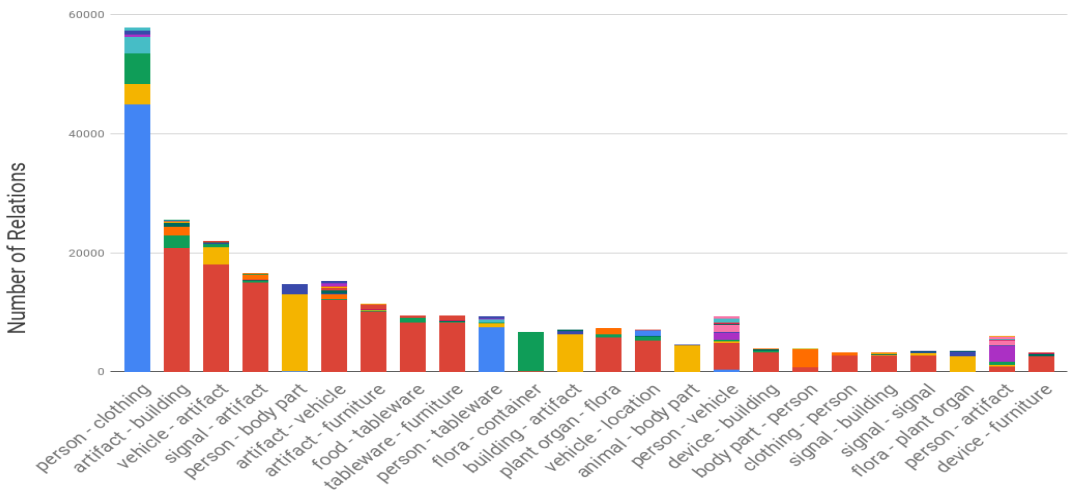


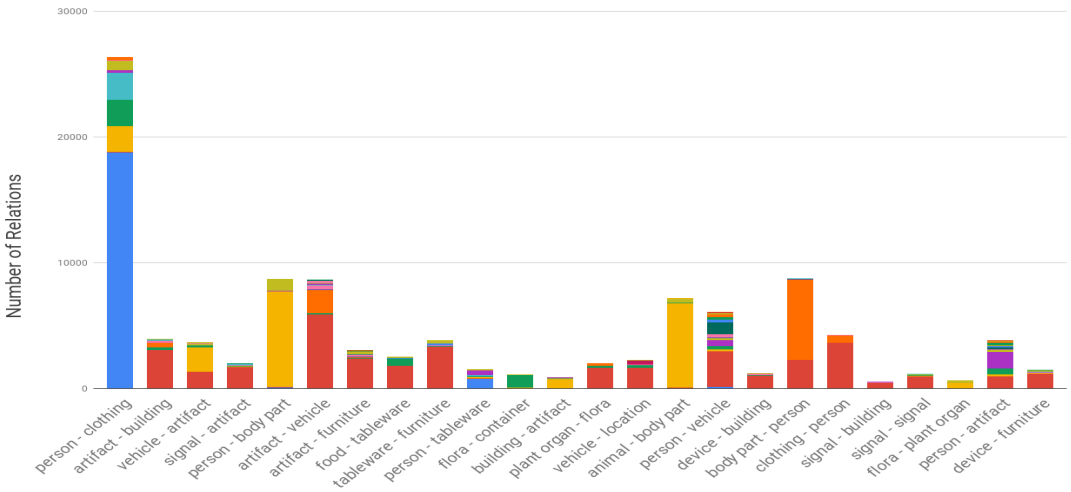
Figure 3.4.2 – Comparison of distributions of relation classes in the test sets of VG-IMP [158] and VG-RMatters (Ours)

The number of test images was reduced so as to keep a high proportion of uncommon relations, so the proportion of images in the split is set at 85%/5%/10% from 70%/5%/25% in [158].

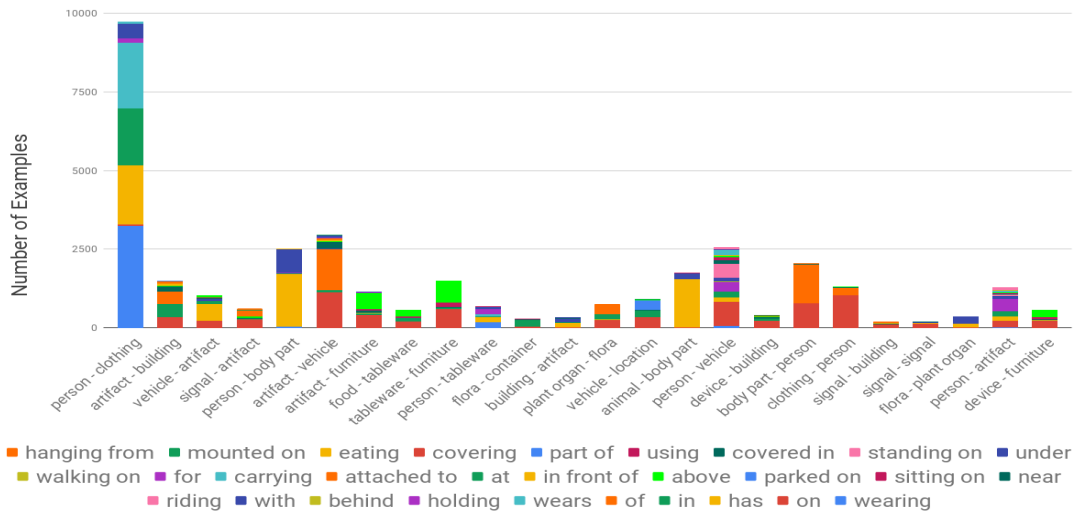
Figure 3.4.3 shows the differences in distribution between both splits for the most frequent category pairs. For all pairs, the proportion held by the majority class is much lower in our split. In Figure 3.4.2, the test distribution is shown independently from object categories for both splits. The proportion of the most common relations decreases in the new split. However relation *on* remains very frequent with 25% of all examples.



(a) Relation train distribution in VG-IMP [158]



(b) Relation test distribution in VG-IMP [158]



(c) Relation test distribution in VG-RMatters (ours)

Figure 3.4.3 – Distribution of relations for each object category on VG-IMP [158] and VG-RMatters

3.4.4 Comparative study

Table 3.4.2 shows performance of three models on both datasets: Freq Baseline predicts the most frequent relation given the class of each object in each pair. MotifNet [166] captures higher order correlations between objects and relationships using LSTM layers. Baseline corresponds to the model shown in Figure 3.3.1.

	SgCls		RelCls					
	ImMacro		ImMacro			ClsMacro		
	R@20	R@100	R@10	R@20	R@100	R@10	R@20	R@100
VG-IMP								
Freq Baseline	31.0	43.9	39.3	52.9	80.0	6.4	11.2	33.7
MotifNet [166]	**37.6**	**47.7**	52.5	66.6	88.3	9.6	15.8	38.1
Baseline (Ours)	37.1	47.2	**52.6**	**66.7**	**88.7**	**10.6**	**17.6**	**41.3**
VG-RMatters								
Freq Baseline	28.4	50.9	25.9	40.7	77.4	7.6	14.2	42.7
MotifNet [166]	38.0	55.5	39.6	58.4	87.8	11.8	19.8	46.6
Baseline (Ours)	**38.5**	**56.3**	**40.3**	**58.9**	**88.3**	**12.6**	**21.1**	**47.8**

Table 3.4.2 – Performance differences between VG-IMP and VG-RMatters. Recalls are in % and evaluated without scene graph constraints.

The differences in performance of FreqBaseline between VG-IMP and VG-RMatters on RelCls, which deteriorates from 39.3% to 25.9% shows that predicting the most frequent relation is a less viable strategy. This is most visible for small scene graphs, on R@10 and R@20, where the performance of all models drops by more than 10 and 8 points respectively. Figure 3.4.4 shows that correct relation is still in the top 2 relations of each given object pair 75% of the time and 87% of the time in the top 3 relations, thus our dataset does not significantly change the scores in the setting of R@100. For reference, there are in average 2.5 relation predictions per object pair for MotifNet. Therefore we focus on the R@10 and R@20 settings which are more demanding considering the structure of example annotations.

Furthermore, we see a better correspondence between high image and class macro recalls, hence it is important to measure this score in order to better understand how the model generalizes.

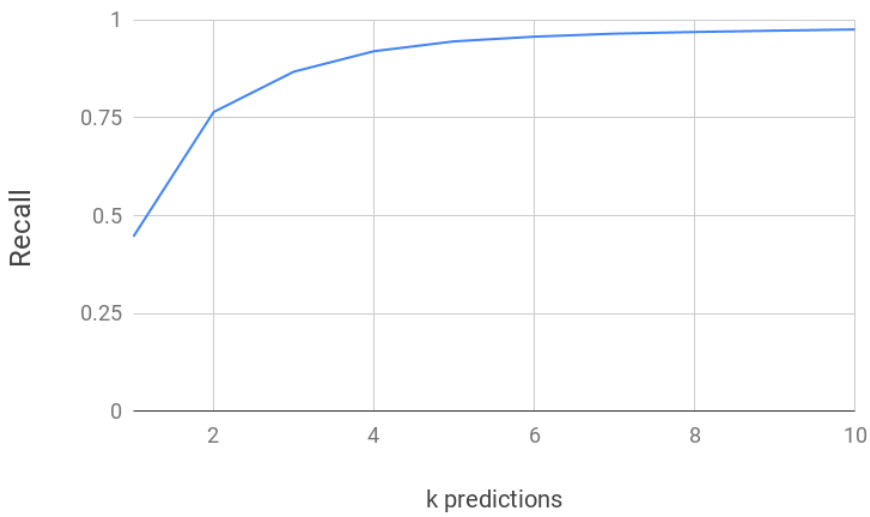


Figure 3.4.4 – Probability of guessing the correct relation in VG-RMatters by predicting the top k relations given an object pair.

3.5 Conclusion

We have shown that Visual Genome exhibits a significant bias, which hinders the training of the model. It biases the model towards the most frequent relations and hinders training due to the resulting high variance of the annotated relation. We have proposed a competitive relation detection baseline with similar performance on the RelCls task on both Micro and Macro recalls when compared to existing state-of-the art methods. This baseline is able to correctly retrieve most common relations but does not learn to detect rarer ones. To reflect this, we proposed to study the macro recall, averaged over each class.

Furthermore, performance on train, validation and test splits at each training epoch shows that the model does not improve in relation detection after the fourth epoch and that its performance on the train split deteriorates. This shows that the loss used to train the model is not directly correlated to the performance on the VRD test setting. Furthermore, performance on the two tasks SGCls and RelCls are also not directly correlated, which motivates us to use the latter as a criterion for early stopping, instead of the former.

Finally, we proposed a new split of Visual Genome designed so that in the test set each object pair has a more evenly balanced distribution of relation classes. We showed that RelCls should be evaluated on smaller scene graphs, as simple baselines perform relatively well on larger ones. With this new setting, being able to reliably detect which relations are true is made more important than in previous settings where predicting common relationships is a more viable solution.

Chapter 4

Overcoming Relation Imbalance with Semantic Modelling

4.1 Motivation

In this chapter, we develop and compare two training methods both aimed at tackling the imbalance of relation data. By this, we aim to improve the estimation of the true relation distribution, and thus decrease the difference between the experimental and true risks. State-of-the-art models are trained with the assumption that relations are mutually exclusive and that available data is sufficient to learn accurate classifiers. Our approach removes the exclusivity assumption and aims to take into account semantic relations between relation classes.

We tackle one consequence of the reporting bias described in 3: the imbalance in the number of example per relation class. This imbalance has strong consequences: it biases predictions towards the most frequent ones, preventing uncommon relations from being learned or retrieved at test time. This is illustrated by the confusion matrix in Figure 4.1.1, which shows that many relations are not learnt and are instead classified as one of the most frequent relations, such as *has, on, in.*

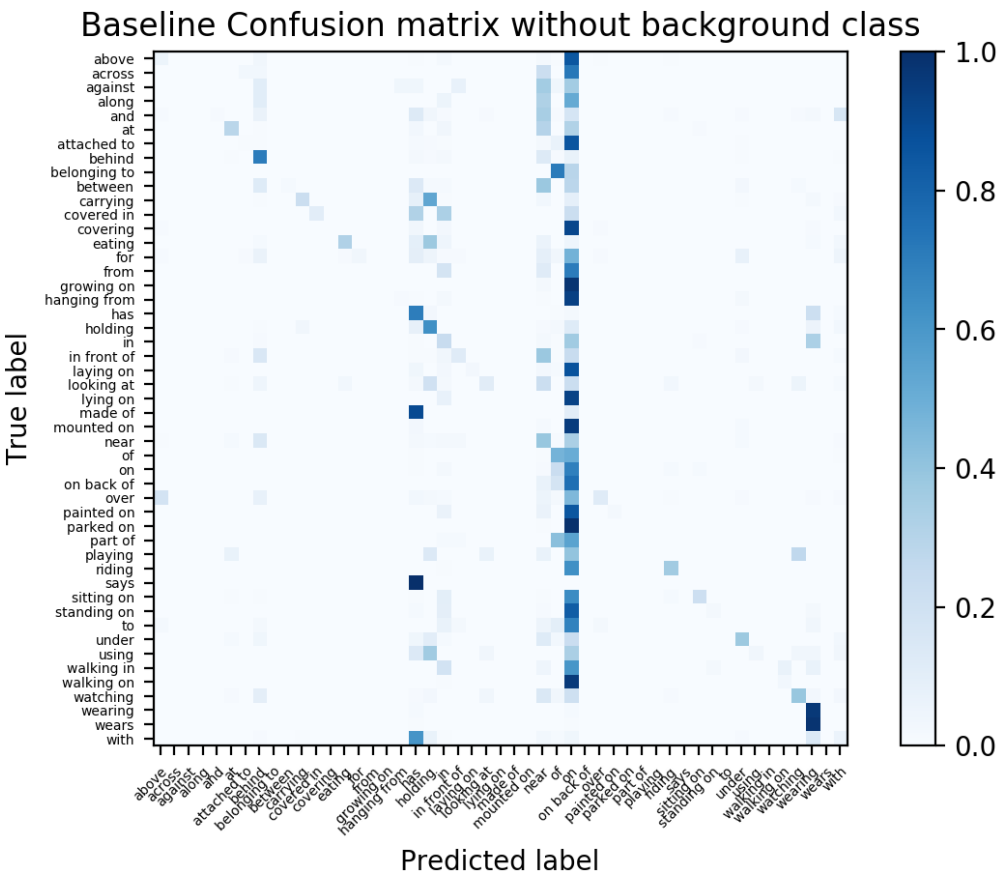


Figure 4.1.1 – Confusion Matrix of our Baseline without the *background* class. Many relations, such as *above, attached to, parked on, walking on*, are rarely detected and are miscategorized as either *has, on* or *in.*

Boundaries between relations are fuzzy as different relations have the same mean-

ing in specific cases. Learning representations where each relation class is separated from other classes requires a high number of examples. This is especially true in the case of relations which describe object pairs at different semantic levels. For example, in Figure 4.1.2, relations *on, above* and *standing on* are true for the selected pair. In order to be able to separate them, it is necessary to have examples in many different situations, so as to successfully separate them when these relations do not overlap.

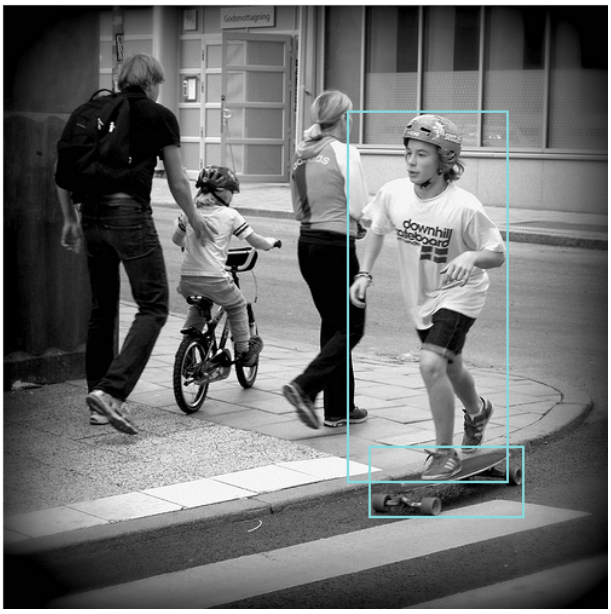


Figure 4.1.2 – Example from Visual Genome. Several true relations describe the elected pair at different semantic levels.

Furthermore, this makes training hard by provoking under-fitting, *i.e.* preventing the model from capturing the high variance of the data. Finally, standard methods for supervised learning are reliant on a high number of examples to be able to learn each class.

To remedy this, we define two methods that alleviate the need in annotations for each class and are able to learn and retrieve rarer classes. These methods differ from each other by how they require external knowledge and how the assumption that relations are exclusive is relaxed.

— The objective is to better learn rarer relations by learning representations aimed at keeping small distances between instances of the same relation classes. This allows the model to embed similar relation classes in the same local spaces, instead of defining hyperplanes that separate the relation space.

— Knowledge is integrated into the model during the training phase by enforcing model outputs to respect a set of constraints. These constraints are defined from external knowledge to quantify how the estimations of a model conform to external knowledge, such as synonymy or semantic similarity. During the

training phase, a loss term is added to the relation classification loss so as to train the model to respect these constraints.

4.2 Learning relation Prototypes

In the next Section, we first present works directly related to modelling the distances between instances, how some of these methods have been adapted in the context of VRD. Second, we propose a method that extends a work by Kaiser *et al.* [65] to the context of classification, relaxing the constraint that relations are exclusive during the training phase. We show that it enables the classification of relations while taking into account their polysemy and synonymy.

4.2.1 Approach extended to VRD

Here we aim to take into account the semantic relationships between relations and thus propose a method robust to the high level of noise in relation annotations. This method, built upon the work of Kaiser *et al.* [65], is adapted to a classification setting, contrary to the few-shot setting of [65]. Consequently, the inference process needs to be different from the trained model very different from the original setting.

This method relies on a small number of relation prototypes. These prototypes are defined such that each relation is summarized by a few references. Using a triplet loss with margin, we train our network to extract representations of relations that capture their diversity of meanings and group together instances of similar meaning, such as *wearing clothes* and *in clothes*. We show that the learnt representation makes use of visual, spatial and semantic information to group together relations with similar meanings, such as *person ride horse* and *person ride elephant*. At test time, relations are classified by using the cosine similarity with the defined prototypes. In the subsequent section, we compare the implicit modelling of synonyms with two methods that adapt the network objective function making use of a set of synonyms. The first is a data augmentation method. The second is inspired by Hu et al. [62]. It constrains the output of the network to a space where synonym detection scores are similar.

4.2.2 Relation representations

This model learns to embed relations in a metric space. These parametric representations are defined as the concatenation of the representations from three modalities: visual, semantic and spatial. We add the semantic representation as the word vectors are defined in a metric space, which makes them useful for our purpose.

For spatial, we change from the previously described representation in order to decrease the number of parameters to learn and use a simpler representation, as the convolutional representation provided worse performance.

Spatial representations We extract the spatial features of the object pair (o_1, o_2) in image I from the bounding box coordinates, as well as the size of the image, as described and shown to be discriminative of relations in [174].

$$\boldsymbol{f_s}(o_1, o_2) = \left[\boldsymbol{f}_{\text{coord}}(o_1, o_2, I), \boldsymbol{f}_{\text{dist}}(o_1, o_2, I), \boldsymbol{f}_{\text{area}}(o_1, o_2, I)\right] \tag{4.1}$$

where

$$\boldsymbol{f}_{\text{coord}}(o_1, o_2, I) = \left[\frac{x_{1,1}}{w_I}, \frac{y_{1,1}}{h_I}, \frac{x_{1,2}}{w_I}, \frac{y_{1,2}}{h_I}, \frac{x_{2,1}}{w_I}, \frac{y_{2,1}}{h_I}, \frac{x_{2,2}}{w_I}, \frac{y_{2,2}}{h_I}, \frac{w_1}{w_2}, \frac{h_1}{h_2}\right] \tag{4.2}$$

$$\boldsymbol{f}_{\text{dist}}(o_1, o_2, I) = \left[\frac{x_{2,1} - x_{1,1}}{w_I}, \frac{y_{2,1} - y_{1,1}}{h_I}, \frac{x_{2,2} - x_{1,2}}{w_I}, \frac{y_{2,2} - y_{1,2}}{h_I}, \frac{x_{2,c} - x_{1,c}}{w_I}, \frac{y_{2,c} - y_{1,c}}{h_I}\right] \tag{4.3}$$

$$\boldsymbol{f}_{\text{area}}(o_1, o_2, I) = \left[\frac{A(o_1)}{A(I)}, \frac{A(o_2)}{A(I)}, \frac{A(o_1)}{A(o_2)}, \frac{A(o_1 \cap o_2)}{A(o_1)}, \frac{A(o_1 \cap o_2)}{A(o_2)}, \frac{A(o_1 \cup o_2)}{A(\text{sup}_{12})}\right] \tag{4.4}$$

$(x_{*,1}, y_{*,1})$ and $(x_{*,2}, y_{*,2})$ are the coordinates of the bottom left and top right corners of the bounding boxes. w_* and h_* denote the width and height of the object bounding box or image. $x_{*,c}$ and $y_{*,c}$ are the coordinates of the center of the bounding box. A denotes the area, $o_1 \cap o_2$ and $o_1 \cup o_2$ are the intersection and union of both bounding boxes and sup_{12} is the smallest bounding box enclosing both o_1 and o_2.

Thus, we make use of three spatial inputs: $\boldsymbol{f}_{\text{coord}}$ for the positions and dimensions of the bounding boxes, $\boldsymbol{f}_{\text{dist}}$ for the distances between objects and $\boldsymbol{f}_{\text{area}}$ for the relative areas of objects, and how much they overlap.

Semantic Representation For the semantic inputs, we use pre-trained word vectors to represent ground truth object classes. The visual and semantic feature vectors [97] are concatenated together with the spatial features and passed through a new fully connected layer in order to get a single representation of the object pair.

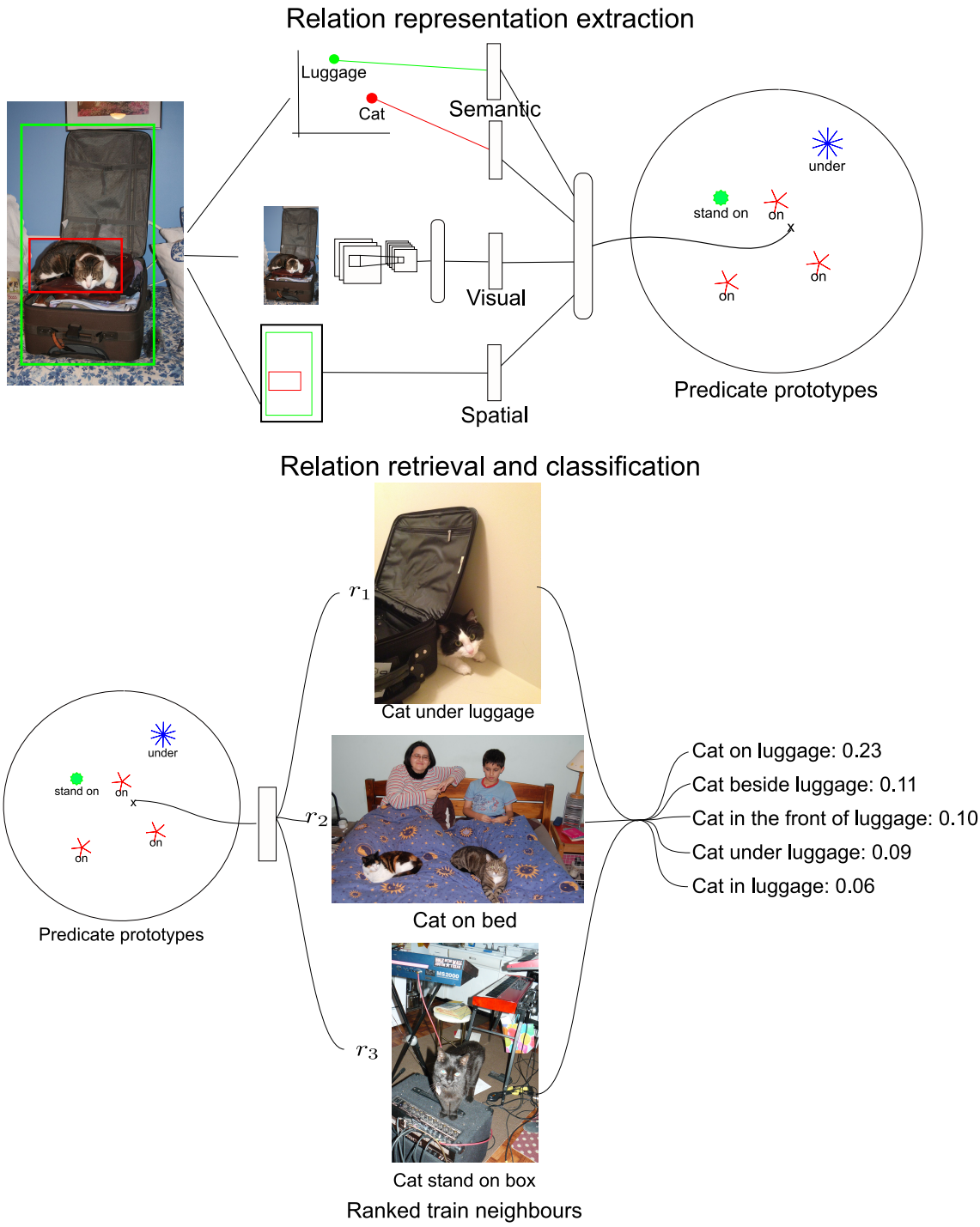


Figure 4.2.1 – Processing pipeline of our model at test time. Visual, semantic and spatial feature vectors are extracted and concatenated and the dimension is reduced using a single fully connected layer. The relation is represented in a space populated by relation prototypes. Then, the closest prototypes are retrieved and the corresponding relation triplets are combined to produce a probability distribution over the space of relations.

4.2.3 Learning Prototypes

To learn relation representations, inspired by [65], we define $\mathcal{M} = \{\mu_1, \ldots, \mu_n\}$, a set of n prototypes where each prototype μ_i is a vector of dimension d and is associated to a relation v_i. During the training phase, $\mathcal{M}$ is updated so that, for each batch and each training sample x of the batch:

$$\arg\min_{\mu \in \mathcal{M}} \|x - \mu\| \in \mathcal{M}_r \tag{4.5}$$

where r is the relation class associated to x and $\mathcal{M}_r$ is the set of prototypes of class r. Each relation may have several different prototypes to allow for the possible differences in visual representations and the polysemy of the relation (e.g. "*in*" describes different interactions in (*man, in, shirt*) and (*man, in, car*)). To allow for synonyms between relations, we relax equation (4.5). Instead of the closest prototype, we enforce that at least one of the p *nearest prototypes* of x has class r (p is determined experimentally). Prototypes are sorted by decreasing cosine similarity to x: $\{\mu_{i_1}, \ldots, \mu_{i_n}\}$ and their associated relations are $\{r_{i_1}, \ldots, r_{i_n}\}$. Let i_{pos} and i_{neg} the smallest indices, such that $r_{i_{pos}} = r$ and $r_{i_{neg}} \neq r$. The loss function is

$$\mathcal{L}(x, r, \mathcal{M}) = \left[x \cdot \mu_{i_{neg}} - x \cdot \mu_{i_{pos}} + \alpha\right]_+ \tag{4.6}$$

where $[.]_+ = \max(., 0)$ and $\alpha > 0$ is a margin, beyond which the distance between both prototypes is large enough.

Prototype updates During the training phase, if a sample does not verify Equation 4.5, an existing prototype is replaced with this sample, in order to keep the same number of prototypes. We select prototypes that have been updated the least recently, as they are more likely to be outliers and are associated to a low number of training samples. Thus, each prototype is associated to an age value $(a_1, \ldots a_n)$ that is incremented during each iteration. This value is updated based on the values of $\{r_{i_1}, \ldots, r_{i_p}\}$, the classes of the prototypes closest to x.

— Case $i_{pos} \in \{1, \ldots, p\}$: the closest prototype of class r is in the p nearest prototypes, then we simply update the vector representation of the prototype:

$$\mu_{i_{pos}} \leftarrow \frac{\mu_{i_{pos}} + x}{\|\mu_{i_{pos}} + x\|}, a_{i_{pos}} \leftarrow 0 \tag{4.7}$$

— Case $i_{pos} \notin \{1, \ldots, p\}$: a new prototype needs to be added to represent x. Here x is added to the set of prototypes and μ_i is updated:

$$\mu_i \leftarrow x, r_i \leftarrow r, a_i \leftarrow 0 \tag{4.8}$$

where $i = \arg\max_{i \in \{1,\ldots,n\}} a_i$.

Learnt prototypes To further explore the learnt representation space, we perform a k-Means [91] on the set of prototypes and for each cluster, we list the (*subject, relation, object*) triplets corresponding to each point. Some clusters are highly consistent both in the concepts for the subject and object role as well as the relations, which are synonyms. One such cluster is comprised of only *(person/animal, clothing)* pairs with relations *in, wear, with, hold* or *has* as shown in Figure 4.2.2. Figure 4.2.3 shows a cluster where relations all entail a close proximity between subject and object, often implying that the subject belongs to the object (*on, of, with, wear, for, belong*) with more varied subjects and objects.

(a) Subject classes (b) Relation classes (c) Object classes

Figure 4.2.2 – Word cloud of a cluster with relations (*person, clothing*) pairs.

(a) Subject classes (b) Relation classes (c) Object classes

Figure 4.2.3 – Word cloud of a cluster with relations entailing close positions, especially possession: on, of, for, wear, belong, part...

(a) Subject classes (b) Relation classes (c) Object classes

Figure 4.2.4 – Word clouds a cluster with (*animal/person, flat surface*) pairs.

This suggests that these clusters could be used to extract synonymy relations be-

tween relation classes when involved with the type of objects of the cluster. However, we also denote two other types of clusters:

1. Comprised of subjects and objects with bounding boxes of similar shapes and in similar configurations. Thus, the visual relations are similar, but the text relations have very different meanings. For example in the cluster with the classes in Figure 4.2.4, with triplets such as *(person, stand on, beach)* and *(zebra, cross, road)*.
2. Comprised of either very semantically similar subjects (resp. objects), with varying objects (resp. subjects) and relations, such as *(sign, in front of, wall)* and *(umbrella, behind, wall)*.

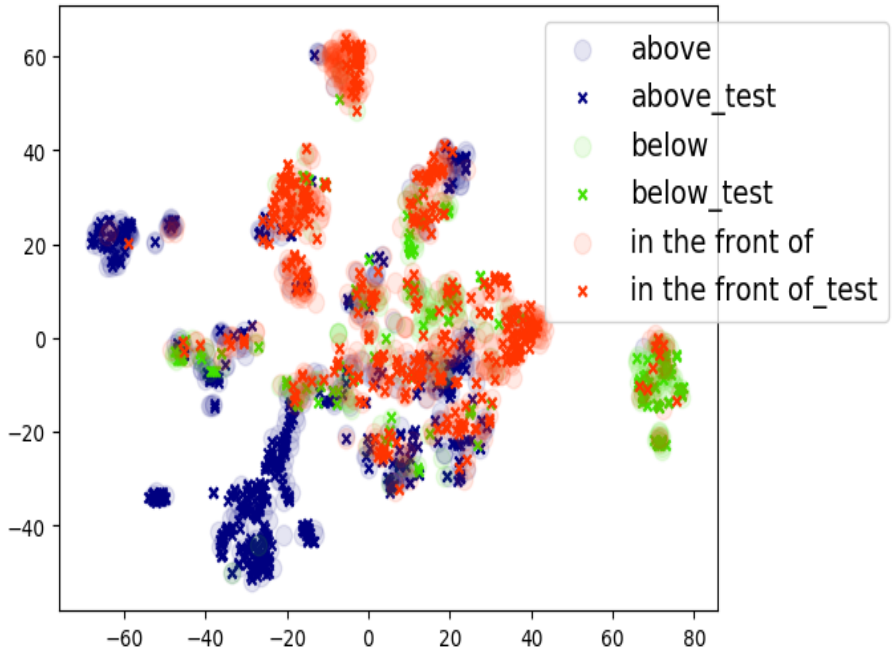
This observation suggests that the similarities of visual, semantic and spatial are taken into account. However, allowing triplets with different relations to be close not only takes into account the synonymy of relation but tends to also decrease the separability of relations classes involving the same pairs of objects. Indeed, when some object pairs can have multiple different relations, such as *person* and *skateboard* and unlike *person* and *hat*, they will have similar visual and spatial features and the model will not be penalized when the closest relation to the relation *(person, carry, surfboard)* is *on*. Furthermore, this suggests that the distance between semantic, spatial and visual features is not sufficient to learn that some relations are common for a given object pair (e.g. cross for (*zebra, road*)) but not for another similar pair (*person, beach*)).

Finally, these results highlight the trade-off that comes with introducing the visual and semantic features of both objects. This helps the model learn correlations between types of objects and relations that are common for these objects, such as *ride* or *wear* and thus introduces a form of common-sense knowledge. However this also increases the dimension of the space of representation and makes it harder to determine which features can really separate relations with high intra-class diversity, such as *on* and *near*.

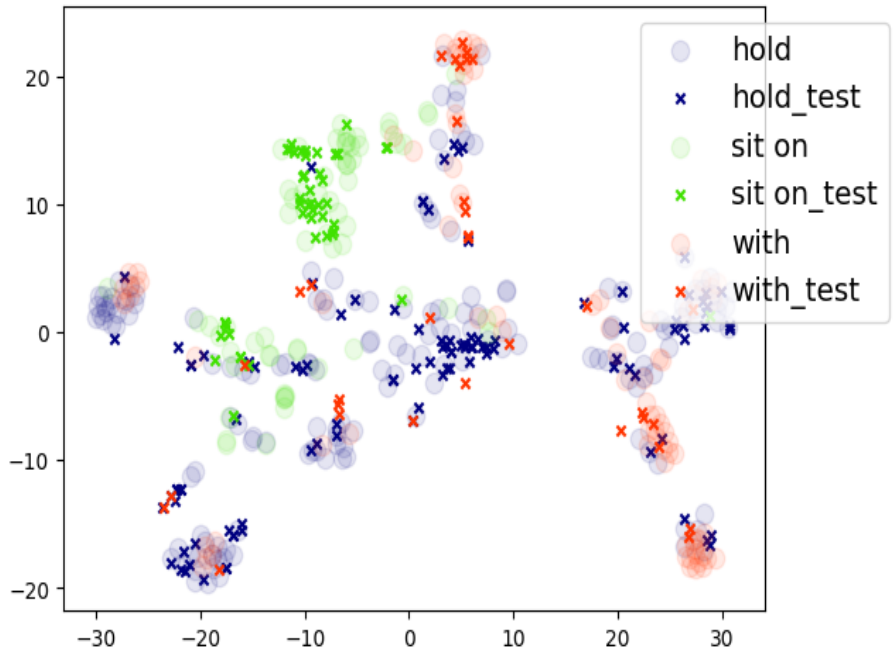
Figure 4.2.5 shows the distribution of relation triplets by relation classes. First, our model is able to group instances of the same relation class, especially for well separated spatial relations, such as in Figure 4.2.5a where we observe some well defined clusters. However the pair (*below, in front of*) is not well separated. Since spatial relations depend on the perspective of the photographer and objects, the modes of different classes can coincide (e.g. *in front of* translates into very different configurations depending on where a person looks), thus these classes are harder to separate. Indeed, we find that on VRD-set, spatial relations have a lower recall than actions (respectively 79% and 88%).

Finally, in Figure 4.2.6, *in* and *wear* have well defined clusters and also three classes share two common cluster, which shows that our model learns to cluster

semantically close examples, such as *(person, relation, clothes)* triplets. In all cases, the distribution of test examples coincides with that of prototypes, which shows that the model successfully learns a generalizable representation of the relation and stores prototypes that correctly represent the class.



(a) Spatial relations



(b) Diverse semantic relations

Figure 4.2.5 – t-SNE [141] embedding of relation representations learnt by *ProtoNN* for relation classes of VRD-set. Crosses correspond to test relations and circles (less opaque) to prototypes stored during the training phase. Our model is able to group instances of the same relation class, especially for well separated spatial relations such as *above* and *below*. For most relations, we notice well defined clusters but also mixed clusters. This suggests that the model learns not only to group semantically close relations (*e.g.* *hold* and *with*), but also relations with visually close relations, which hinders performance.

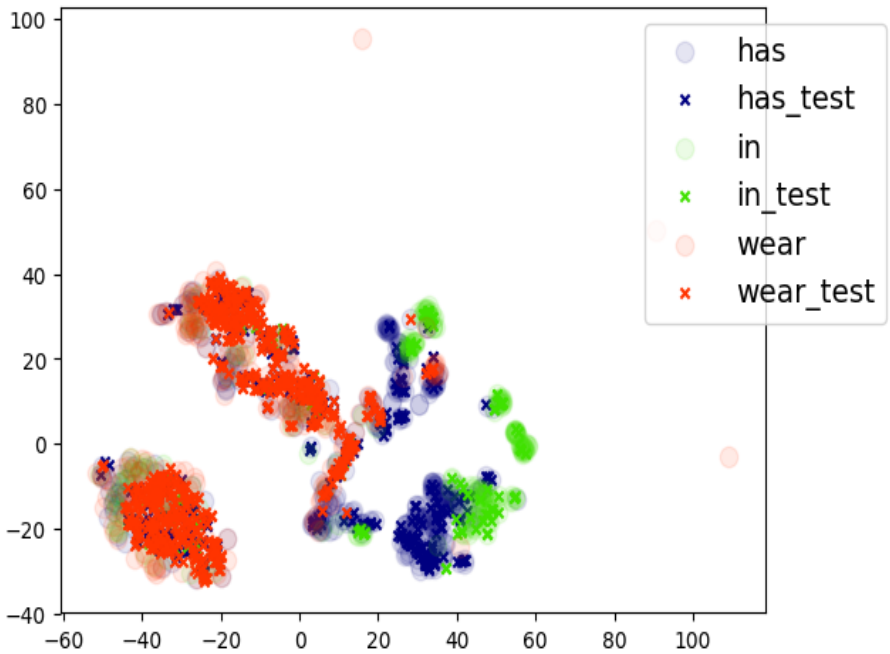


Figure 4.2.6 – t-SNE [141] embedding of relation representations learnt by *ProtoNN* for relation classes of VRD-set. Crosses correspond to test relations and circles (less opaque) to prototypes stored during the training phase. Our model learns to cluster semantically close examples, such as *(person, relation, clothes)* triplets.

Figure 4.2.7 – Examples of retrieved nearest neighbors for four test examples. For the top 2 examples, our model captures semantic similarities between objects (e.g. *horse* and *bear*) and between spatial configurations (top retrieved relations are very different for *ride* and *next to*). Third example shows a failure case. Fourth example shows an example of correct classification where the closest relation is incorrect but the correct relation is retrieved by voting.

Figure 4.2.7 shows that word vectors play an important role in the generalization capabilities of our model as it learns to cluster relations with semantically similar objects. On the first line, the closest relations to $(person, horse)$ involve the pair $(person, bear)$ even though the bear and horse are visually different. The second line represents a relation between objects similar to those of the first line. However the retrieved relations are different to those of the first due to differing visual and spatial information. The third example shows a failure case where the closest neighbours have similar spatial relations and objects are semantically similar. Due to the rarity of relation "*touch*", the correct relation is not retrieved. Fourth example shows an example of correct classification where the closest relation (with the most similar spatial configuration) is incorrect but the correct relation is retrieved by voting.

4.2.4 Inference

In the following, we compare performance on two different inference strategies. The first is based on comparison with prototypes, similarly to how Kaiser *et al.* [65] made use of memory keys. The second relies on the comparison with previously seen training samples, which has the advantage of also allowing the comparison with object classes from these examples.

Instance to prototype comparison At test time, the probability of relation r_k being true for an object pair (h, t) is computed as:

$$p(e_{h \to t,k}) \propto \sum_{i \in [1,n] | r_i = r} x \cdot \mu_i \tag{4.9}$$

where x is the representation of the object pair (h, t).

Instance to instance comparison After the training phase described in Section 4.2, we store the representations of all annotated train relations. At test time, given image I, the learnt representation of all object pairs is extracted using ground truth object bounding boxes and classes. For each pair, the top k nearest neighbors from the train set are retrieved with their distance to the query, as illustrated in Figure 4.2.1. The similarity between each neighbor x_i, $i \in \{1, \ldots, k\}$ and the query q is computed with a softmax function:

$$\text{sim}(q, x_i) = \frac{e^{-d(q,x_i)^2}}{\sum_{j=1}^{k} e^{-d(q,x_j)^2}} \tag{4.10}$$

Semantic consistency. To increase the consistency of the predictions, the similarity is multiplied by a semantic consistency factor: $\text{consistency}(q, x_i) = \text{sem}(s_q, s_{x_i}) \cdot$

$\text{sem}(o_q, o_{x_i})$ where h_q, h_{x_i} are the classes of the head objects, t_q, t_{x_i} are the classes of the tail objects and $\text{sem}(.,.)$ is the scalar product of word embeddings [97]. Finally the weight associated to relation r_i is:

$$w_i = \frac{\text{sim}(q, x_i) \cdot \text{consistency}(q, r_i)}{\sum_{j=1}^{k} \text{sim}(q, x_j) \cdot \text{consistency}(q, x_j)} \tag{4.11}$$

Thus the probability of relation v being true is the sum of the weights of all relations annotated with relation v.

4.2.5 Experiments

Implementation details The maximum number of relation prototypes n, which are represented in a $d = 512$ dimensional space, is set to 8192 as in [65], which enables network training convergence and keeps at least one prototype of each class. p is experimentally set to 5, as we found that performance is not improved beyond that. To train the network, the sum of the triplet loss (Section 4.2) and the cross-entropy loss for object classification is optimized with an Adam optimizer with learning rate 1.10^{-3}. Finally, on VG, we use an approximate similarity search [63] in order to speed up computations.

Comparative study We compare our method to state of the art methods on the VRD-set and VG datasets on the RelCls task. DSR - SR [83] is trained with a Structural Ranking loss to increase the relevance of predicted annotations and DSR - CE with cross-entropy. KD [165] is a network trained with a loss function augmented with knowledge from the dataset statistics, and KD - no stat is the same network trained without this knowledge. Pixel2Graph [102] is a recent scene graph prediction model that iteratively refines object and relation heatmaps. DG [158] iteratively refines relation representations using a Message Passing framework, without using spatial features as input. Zoom-Net [164] integrates a message passing framework between objects and object pair regions at a lower representation level and makes use of a concept hierarchy to train the network to produce semantically consistent predictions.

Furthermore, a relation classifier Baseline is trained with a cross-entropy loss using all inputs. ProtoNN is a network trained with a prototype triplet loss that classifies relations based on the distance to the closest prototypes. TrainNN is a classifier based on the distance to the nearest train relations, trained with cross-entropy CE or prototype triplet loss PL. For the Baseline, the relation representation is the concatenation of the output of the fc8 layer, word vectors and spatial features. This baseline provides poorer performance than the network shown in

Chapter 3, because of the absence of spatial convolutional representation and the differences in negative example sampling. Therefore, we focus our study on the comparison between baseline and prototype learning. Further comparison with an updated baseline would be required to provide further proof.

	R@50	R@100
KD - NO STAT [165]	72.3	84.9
KD [165]	85.6	94.7
DSR - CE [83]	79.2	89.2
DSR - SR [83]	86.0	93.2
ZOOM-NET [164]	**89.0**	**94.6**
BASELINE (Ours)	77.0	86.3
PROTONN (Ours)	73.1	83.9
TRAINNN - CE (Ours)	69.8	79.4
TRAINNN - PL (Ours)	78.6	87.6

Table 4.2.1 – RELCLS results on VRD-set. Classification with prototypes has limitations that do not appear in the one-shot setup of [65]. Classification with Nearest Neighbors and prototypes (TRAINNN-PL) outperforms our BASELINE but is outperformed by several state of the art methods.

The object detection system is supposed to be perfect thus we used the word embeddings of the ground truth object classes at test time. This however makes the comparison with some state of the art methods more difficult.

Classification with prototypes Table 4.2.1 shows that on VRD-set, BASELINE using all inputs outperforms PROTONN. Classification with prototypes has limitations that do not appear in the one-shot setup of [65], where the classification is done by comparing the query to a small number of examples of various classes, whereas here the query is compared to a high number of prototypes to overcome the problem of synonymy between relations.

Classification with nearest neighbors To overcome this, we rely on the annotations of the training set as described in Section 4.2.4. Retrieving a high number of neighbors increases the bias of our classifier. The classification setup with nearest neighbor TRAINNN and BASELINE is largely outperformed by the same network with triplet loss TRAINNN - PL, as the former is not trained to group representations of relations but to define a space where classes are linearly separable. TRAINNN - PL obtains better performance on datasets where train and test distributions are similar.

	R@50	R@100
DG [158]	45.3	58.2
DSR - SR [83]	69.1	74.3
PIXEL2GRAPH [102]	**82.0**	86.4
SGP [58]	80.8	88.2
MOTIFNET [166]	81.1	**88.3**
BASELINE (Ours)	73.9	82.5
PROTONN (Ours)	72.2	81.5
TRAINNN - PL (Ours)	67.9	76.0

Table 4.2.2 – RELCLS results on VG-IMP. Classification with prototypes shows poorer performance on VG-IMP than our BASELINE, which highlights the limitations of our representation. Relation representations group together visually and spatially similar relationships but these do not necessary correspond to the same relation.

4.2.6 Discussion

Table 4.2.2 shows that on VG-IMP, PROTONN outperforms PIXEL2GRAPH [102] with a recall of 81.4% but is outperformed by the more recent works. Here PROTONN is outperformed by BASELINE (R@100 = 82.5%) but outperforms *ProtoNN - PL* (R@100 = 76.0%). Furthermore, the study of prototype representations highlights the trade-off between relying on visual and semantic representations to predict relations among a set of possible relations and relying on spatial information to predict relations outside the most common relations. Indeed, the network learns which features correlate the most with each relation. However, visual and semantic features may highly correlate to a set of relations without this relation being true, due to the spatial configuration of objects. Due to the high imbalance in relation class for each object pair and to the similarity metric which weighs each feature equally, the model is not able to correctly separate them.

4.2.7 Conclusion

We presented a new visual relation detection model that embeds relation instances into a space populated with relation prototypes, where similar relations are clustered together. Results show that prototype learning is very promising, when it comes to predicting a larger variety of relations while maintaining a high average recall. However, the classification process at test time requires further exploration. Indeed our setting is not able to reliably classify relations among a higher number of classes than the few-shot learning setting. The structure of the learnt space shows that the model learns to cluster semantically close relations, and stores prototypes that correctly represent relation classes but that the defined clusters also define

clusters of object pairs, which hinder relation classification.

4.3 Learning rarer classes with External Knowledge

In the previous section, we modelled relation similarity by learning a metric space populated with prototypes and by enforcing that object pairs encountered during training are close to a prototype of the same relation class. The model learns to group similar relations. However, the similarity is impacted if relation instances share similar objects which biases the model as shown in Figure 4.2.4.

In the following sections, we investigate the impact of training a model to explicitly consider the synonymy between relations classes and their similarities. We classify these methods as **Explicit Semantic Modelling** (which rely on external semantic data), as opposed to the prototype learning method described in Section 4.2, classified as **Implicit Semantic Modelling**. Furthermore, we compare two different methods of modelling synonymy between relations: data augmentation with synonyms (Section 4.3) and rule distillation (Section 4.3.4).

4.3.1 Related Work

In [40], Galleguillos *et al.* show that knowledge, in the form of statistics measured on the studied dataset, can be integrated into the formulation of a Conditional Random Field. This makes outputs more consistent with the whole set of training samples and thus increases its predictive power. Knowledge may also be integrated during the training phase of a model. Rohrbach et al. [123] show that external knowledge on attributes enables the detection of classes without training samples. They connect classes with attributes and use attributes common to several classes to recognize instances of unseen classes. Specifically on Visual Relation Detection, knowledge has been integrated in the form of semantic modelling of relations [89, 117].

External knowledge may be integrated into neural networks using rule distillation as shown by [62] (see Section 2.4.3) and more recently by [165], where internal and external knowledge are used to improve relation classification. However, one drawback of [165] is that it is not possible to extract significant knowledge for each relation when considering a context of several thousand classes, which we aim to tackle here. To overcome such limitations, we introduce a different semantic rule distillation scheme that is capable of treating a wider range of classes and increases scalability by limiting the burden of directly using large external corpora.

4.3.2 Sources of External Knowledge

In this section, we explore the use of two different kinds of knowledge. The first one is a set of synonyms, describing the fact that a true relation implies that several other relations are true. The second one quantifies how semantically similar relations tend to occur in similar contexts.

Synonym Dictionaries Let $\mathcal{S} = \{(r_{i1}, r_{j1}), \ldots, (r_{in}, r_{jn})\}$ a dictionary of synonyms, where for each $k \in [1, n]$, $r_{ik}, r_{jk} \in \mathcal{R}$ and $(r_{ik}, r_{jk}) \in \mathcal{S}$ if and only if relation r_{jk} is a synonym of relation r_{ik}. The synonymy relation is not considered symmetrical to conform with the used synonym dictionary.

To define $\mathcal{S}$, we use ConceptNet [137]. ConceptNet ($\mathcal{CN}$) is a set describing relationships between concepts. $\mathcal{S}$ is thus defined such that, for each pair of relation classes r_i, r_j, $(r_i, r_j) \in \mathcal{S}$ iff $(r_i, \text{'Synonym'}, r_j) \in \mathcal{CN}$.

Furthermore, ConceptNet provides a weight associated to each relationship, quantifying the confidence in the relationship (*e.g.* edges from WordNet have a weight of 2 while an assertion by one person in Open Mind Common Sense is associated to 1).

Semantic Similarity Similarly to 4.2.2, embeddings are vector representations of words, computed such that words that frequently appear in the same context have embeddings with cosine similarity close to 1.

4.3.3 Data Augmentation with synonymy-compatible distributions

Here, we aim to increase the completeness of annotations in a usual supervised classification setup, where the goal is to obtain a maximum likelihood estimator. Indeed, several annotations can be inferred from an existing relation using synonym dictionaries.

Thus, for $\mathcal{D}$ a set of images, where for each $I \in \mathcal{D}$ with n_I objects and $R_I = \{(o_h, r_{h\to t}, o_t) | h \neq t \in n_I\}$, the set of true relations in I. To obtain a maximum likelihood estimator, the cross entropy between the ground truth distribution p_{data} and the model distribution p_Θ is minimized and the model parameters Θ are defined as:

$$\theta = \arg\max \mathbb{E}_{p_{data}}[-\log p_\theta] \tag{4.12}$$

$$= \arg\max \sum_{I\in\mathcal{D}} \sum_{o_h,o_t\in I} \sum_{r\in\mathcal{R}} -p_{data}(r|o_h, o_t) \log p_\theta(r|o_h, o_t) \tag{4.13}$$

Since the parameters of object and relation detection are separate, we consider only the relation probabilities and the corresponding model parameters. When augmenting data, the parameters are optimized in order to maximize the log likelihood of an augmented data distribution. For $o_h, o_t \in I$ and $r \in \mathcal{R}$, the probability distribution of relations is defined as:

$$p_{data+}(r|o_h, o_t) = \sum_{r' \in \mathcal{R}} \delta_I(r'|o_h, o_t) \cdot f(r, r') \cdot \delta_{\mathcal{S}}(r', r) \tag{4.14}$$

where $f(r, r')$ is the frequency of relation r in the set of all relations synonymous to r':

$$f(r, r') = \frac{n_r}{n_{\mathcal{S}(r')}} \tag{4.15}$$

where n_r is the number of relation triplets with relation r in $\mathcal{D}$ and $n_{\mathcal{S}(r')}$ is the number of relations synonymous to r' in $\mathcal{D}$: $n_{\mathcal{S}(r')} = \sum_{r'' \in \mathcal{S}(r')} n_{r''}$

Let $\delta_{\mathcal{S}}$ the indicator function of $\mathcal{S}$. If the relation triplet (o_h, r', o_t) is true in image I, the function $p_{data+}(.|o_h, o_t)$ is a probability mass function (PMF):

$$\begin{aligned} \sum_{r \in \mathcal{R}} p_{data+}(r|o_h, o_t) &= \sum_{r \in \mathcal{R}} \delta_I(r'|o_h, o_t) \cdot f(r, r') \cdot \delta_{\mathcal{S}}(r', r) \\ &= \delta_I(r'|o_h, o_t) \sum_{r \in \mathcal{S}(r')} \frac{n_r}{n_{\mathcal{S}(r')}} = \delta_I(r'|o_h, o_t) = 1 \end{aligned} \tag{4.16}$$

The underlying intuition is that, in order to generalize to the test examples, the expected frequency for each relation should remain the same as with the maximum likelihood estimator. For r in $\mathcal{R}$

$$\text{freq}_{data+}(r) = \frac{1}{n_{rel}} \mathbb{E}_{p_{data+}}[1_r] = \sum_{I \in \mathcal{D}} \sum_{o_h, o_t \in I} p_{data+}(r|o_h, o_t) \tag{4.17}$$

$$= \sum_{I \in \mathcal{D}} \sum_{o_h, o_t \in I} \sum_{r' \in \mathcal{R}} p_{data}(r|o_h, o_t) \cdot f(r, r') \delta_{\mathcal{S}}(r', r) \tag{4.18}$$

$$= \sum_{I \in \mathcal{D}} \sum_{o_h, o_t \in I} p_{data}(r|o_h, o_t) \tag{4.19}$$

$$= \text{freq}_{data}(r) \tag{4.20}$$

4.3.4 Rule distillation

The previous approach has two major drawbacks. First, it is highly dependent on the quality of the synonym dictionary. Second, relation classes are polysemous, which makes the direct of use of synonyms inadequate. These synonymy relations do not always hold true, depending on the object pair under study. To mitigate these drawbacks, we make use of a rule distillation process, in the same vein as [62].

Instead of changing ground truth annotations, external knowledge is integrated into the network during training by using an additional loss term that quantifies the extent to which a model follows a set of rules.

Definition Let a neural network with parameters θ that outputs a conditional probability distribution $p_\theta(\boldsymbol{Y}|\boldsymbol{X})$ of output variable $\boldsymbol{Y}$ given input variable $\boldsymbol{X}$. As in [62, 165], we define q such that

$$q = \arg\min_{q \in \mathcal{P}} \mathrm{KL}(q(\boldsymbol{Y}|\boldsymbol{X})||p_\theta(\boldsymbol{Y}|\boldsymbol{X})) - \lambda \mathbb{E}_{q(\boldsymbol{Y}|\boldsymbol{X})} f(\boldsymbol{X}, \boldsymbol{Y})) \tag{4.21}$$

where $\mathcal{P}$ is the set of PMF over $\mathcal{R}$ and q is the projection of p_θ on a subspace verifying constraints defined by f. The more $(\boldsymbol{X}, \boldsymbol{Y})$ respects these constraints, the closer $f(\boldsymbol{X}, \boldsymbol{Y})$ is to 1. KL is the Kullback-Leibler divergence. As shown in [62], the closed form solution of Equation 4.21 is $q(\boldsymbol{Y}|\boldsymbol{X}) \propto p(\boldsymbol{Y}|\boldsymbol{X}) e^{\lambda f(\boldsymbol{X}, \boldsymbol{Y})}$.

This new projected probability is added to the original network loss during the training:

$$\begin{aligned} L(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta}) =& (1 - \pi^{(t)}) \cdot l(\boldsymbol{y}, p_\theta(\boldsymbol{Y}|\boldsymbol{x})) + \\ & \pi^{(t)} \cdot l(q(\boldsymbol{Y}|\boldsymbol{x}), p_\theta(\boldsymbol{Y}|\boldsymbol{x})) \end{aligned} \tag{4.22}$$

where $l(\boldsymbol{y}, p_\theta(\boldsymbol{Y}|\boldsymbol{x}))$ is the network loss, corresponding to the cross-entropy between the ground-truth label and the output distribution $p_\theta(\boldsymbol{Y}|\boldsymbol{x})$. $\pi^{(t)}$ is the weight of the distillation loss at iteration t.

At the beginning of the training, since $p_\theta(\boldsymbol{y}|\boldsymbol{x})$ is far from the expected distribution, a large weight on the distillation loss would harm the training process, therefore $\pi^{(t)}$ is set close to 0 and increases during the training phase.

Rule Distillation for VRD Instead of defining rules that the model should follow given the structure of its input, these rules are instead defined based on the expected output. Thus the formulation of 4.21 becomes

$$q = \arg\min_{q \in \mathcal{P}} \mathrm{KL}(q(R_{h \to t}|I)||p_\theta(R_{h \to t}|I)) - \lambda \mathbb{E}_{q(R_{h \to t}|I)}[f(R_{h \to t}, \tilde{R}_{h \to t})] \tag{4.23}$$

where $f : \mathcal{R}^2 \to \mathbb{R}$ is a function quantifying the degree to which a constraint is respected. $R_{h \to t}$ is the relation output variable and $\tilde{R}_{h \to t}$ is the target relation. Thus q is defined such that, the relation classes r with high values of $f(r, \tilde{r})$ have a high probability $q(R_{h \to t} = r|I)$.

Given the closed form solution of this equation, we model f as

$$f(r, \tilde{r}) = \log P_K(r|\tilde{r}) \tag{4.24}$$

where $P_K(r|\tilde{(r)})$ is a the probability that r is true given that $\tilde{r}$ is, according to some external knowledge K. Thus, since $q(R|I) \propto p_\theta(R|I) \exp \lambda f(R, \tilde{R})$, we have

$$q(R|I) \propto p_\theta(R|I) P_K(R|\tilde{R}) \tag{4.25}$$

Synonymy Distillation Using the synonym dictionary, we make the hypothesis that the probability that relation r is true depends on the weight of the synonymy relationship in $\mathcal{CN}$. We aim in this section to reward the model when synonymous relations have similar probabilities. For $I \in \mathcal{D}$, o_h, o_t in I and $r, \tilde{r} \in \mathcal{R}$, such that $(o_h, \tilde{r}, o_t) \in I$, we define P_k as

$$P_K(r|\tilde{r}) = \text{softmax}_{\mathcal{S}(\tilde{r})}(T * w_{r,\tilde{r}}) \tag{4.26}$$

where $w_{r,\tilde{r}}$ is the weight of the synonym between r and $\tilde{r}$, given by the synonym dictionary, $\mathcal{S}(\tilde{r})$ is the set of synonyms of relation $\tilde{r}$ and T is a temperature parameter. This formulation allows us to set the influence of the weight levels on the constraints. The lower T is, the higher the entropy and the closer the distribution is to a uniform distribution. T is set to 2, allowing an object pair to have between 1 and 5 probable relations (i.e. $P(r|\tilde{r}) \geq 0.1$). However this might not be an optimal value, and can be done through a parameter search, which we leave for further research.

Relation semantic similarity Here, we aim to use the knowledge of relation semantic similarities by rewarding the model when semantically close relations have similar probabilities. We characterize the semantic similarity between relations by the cosine similarity between their word vectors. We use the GloVe embeddings defined in [111]. Since these vectors are trained such that the dot product is equal to the log probability of co-occurence, relations that are semantically similar will be relations that appear in similar contexts. Hence in a given context, i.e. a given object pair, the probabilities of different relations to describe this pair are related to their semantic similarity.

Similarly to synonymy distillation, we model $P_K(r|\tilde{r})$ by

$$P_K(r|\tilde{r}) = \text{softmax}(T * \text{sim}(\text{r}, \tilde{\text{r}})) \tag{4.27}$$

where $\text{sim}(r, \tilde{r})$ is the cosine similarity between embeddings of r and r'.

As illustrated in Fig. 4.3.1, the projected distribution has increased probabilities for the relations closest to the ground truth relation and inversely for further relations. This formulation differs from the constraints expressed in [62] as we use the ground truth value to project the output distribution. The loss gradient would be less stable if it was based on the output relation instead of the ground truth.

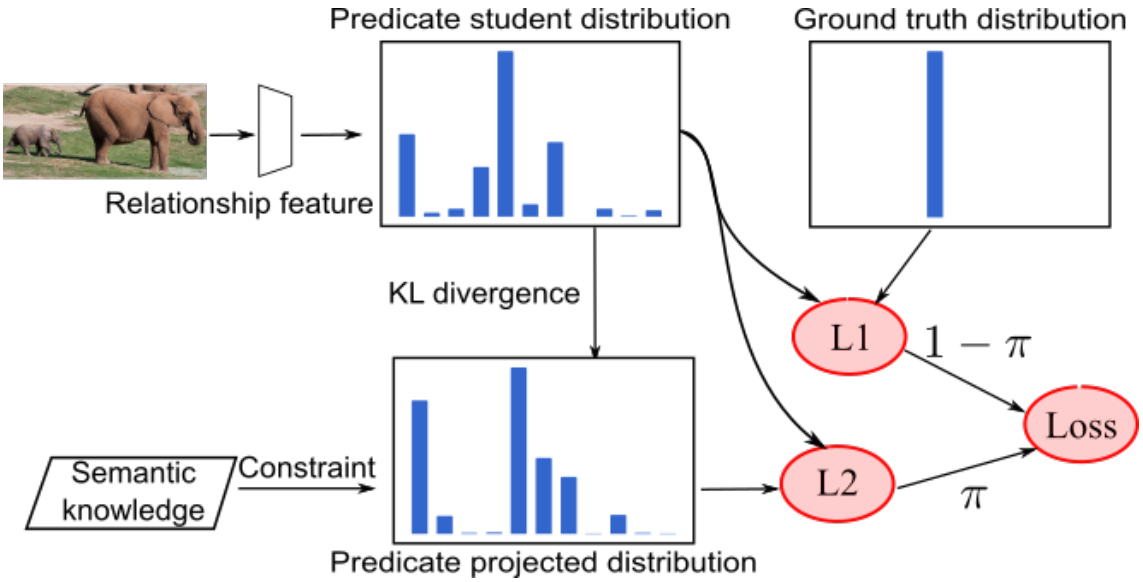


Figure 4.3.1 – Semantic knowledge is distilled into the network by projecting the output distribution under the constraint that relations semantically similar to the selected relation have high probabilities.

Furthermore, it also reduces computations as the constraints can be computed beforehand.

Internal knowledge distillation Finally, we compare the previous distillations with internal knowledge distillation, presented by Yu *et al.* [165]. The purpose of their method is to restrict the outputs to a subset of relations that are the most probable for a given pair of objects.

$$f(R_{h\to t}, I) = \log P_{data}(R_{h\to t}|V_h, V_t) \tag{4.28}$$

where f is the constraint function and $P_{data}(R|V_h, V_t)$ is the pre-computed distribution of relations given pairs of object classes.

Similarly to Section 4.23, the goal is to reward relations frequently associated with the current context. For semantic knowledge distillation, this context was given by the annotated relation. Besides, relations were represented by vectors precomputed over a large text corpus, requiring fewer annotations but missing the specificities of image contexts. For internal distillation, the context is given by the object pair and since the relation distribution is computed on the training annotation set, it is more accurate but requires more data to cover the whole space.

4.3.5 Experiments

We evaluate the performance of our model for relation classification on two datasets. Several setups are compared and show that representations learnt by our model using triplet loss give an improvement over a classical classification scheme.

Implementation details The set of synonyms $\mathcal{S}$ and the associated weights used in Equation 4.26 come from ConceptNet [137], a semantic network with more than 8 million nodes and 21 million edges, aggregating several common sources of knowledge: Commonsense Computing projects, contributors to Wikimedia projects, Games with a Purpose, Princeton University's WordNet, DBPedia, OpenCyc, and Umbel.

For synonym knowledge distillation, both losses are weighed as in [62], where the $\pi(t)$ increases during the training from 0 to 0.05. Other intervals for π were tested: $[0, 0.15], [0, 0.25], [0, 0.5]$ and performance remains stable in this interval. The temperature parameter in equation 4.26 is set to 2.

	RelCls		
	ImMacro		ClsMacro
	R@50	R@100	R@100
VRD-set			
Baseline (Ours)	**77.0**	**86.3**	40.3
SynAugment (Ours)	73.9	83.1	24.5
SynDistill (Ours)	74.0	83.5	34.6
ProtoNN (Ours)	73.1	83.9	**41.7**
VG-IMP			
Baseline (Ours)	**73.9**	**82.5**	31.4
SynAugment (Ours)	70.7	79.1	28.0
SynDistill (Ours)	73.2	82.2	29.9
ProtoNN (Ours)	72.2	81.5	**35.4**

Table 4.3.1 – Results of Explicit and Implicit models of synonyms on VRD-set and VG-IMP. ProtoNN is outperformed by our Baseline for ImMacro but outperfoms all methods for ClsMacro. This suggests that keeping at least one prototype for each relation and modelling their relationships increases the recall of rarer relations.

Explicit and implicit synonym modelling Table 4.3.1 shows the impact of the proposed synonym modelling methods on both datasets. SynAugment refers to the network trained with data augmentation presented in Section 4.3.3. SynDistill refers to synonym data augmentation (see Section 4.3.4). Baseline is a simpler baseline than the one presented in Chapter 3, with only 1 stream and without

spatial convolutions. We notice that the same tendencies hold from one dataset to another. In both, SynAugment has the lowest performance in micro recall, with 83.1% in VRD-set, compared to CE at 86.3%. On VG, it reache 79.1% recall, 3 points lower than Baseline at 82.6%. For macro recall, the results of both methods that explicitly model synonymy, that is SynAugment and SynDistill, show that completing the existing annotations with synonyms hurts the average relation recall.

Semantic Similarity for a large number of classes To further show the impact of our setup for semantic similarity distillation, we study results on a larger dataset: VG-Large. It is defined by taking all examples from object and relation classes with two or more examples. This version contains 20,000 object classes, 10,000 relation classes and 1.8 million relation annotations, 2.5 times more relations than VG-IMP and VG-RMatters For both VG-Large and VG-IMP, we use the training and test split defined by Xu et al. [158].

IK+ stands for internal knowledge distillation [165] and **SK** for semantic knowledge distillation. We note that for **IK**+, to tackle the challenge of the long tail distribution of object classes, we regroup them by common words using a parser and a context free grammar to get the head word of a noun phrase.

On the VG-Large dataset (Table 4.3.2) both semantic similarity and internal distillations bring significant improvements to the prediction task. A 10.5% and 10.7% increase of the R@100 over the original dual graph network is observed, corresponding to 32.1% and 32.7% relative gains. Both distillations bring similar improvements overall. Internal distillation gives slightly better results on the filtered VG and VRD-set while semantic distillation outperforms the latter on Large VG. This shows the value of the presented semantic knowledge distillation, which incorporates knowledge from precomputed word representations into the neural network and can easily be applied to other benchmarks without requiring any additional data.

However results with both distillations combined provide smaller improvements than

	SGCls		RelCls	
	R@50	R@100	R@50	R@100
Dual Graph [158]	8.6	11.2	22.7	32.7
IK+	9.8	12.6	33.1	43.2
SK (Ours)	**9.8**	**12.6**	**33.3**	**43.4**
SK - IK (Ours)	9.9	12.7	33.0	43.0

Table 4.3.2 – Results of Semantic and Internal Distillations on VG-Large. Both distillations significantly improve the training of the model in the case of a large number of classes.

	SGCls		RelCls	
	R@50	R@100	R@50	R@100
Region model [174]	*N/A*	*N/A*	51.5	51.5
IMP [158]	34.6	41.9	60.9	72.6
IK+	**43.5**	**50.5**	**71.3**	**81.9**
SK (Ours)	41.2	48.7	71.0	80.8

Table 4.3.3 – Results of Semantic and Internal Distillations on ImMacro on VRD-set. In the case of fewer relation classes, internal distillation defines rules that generalize better than semantic distillation.

either separately. This might come from the added constraints, which result in a biased estimation of the relation distribution.

Results on VRD-set (Table 4.3.3) show that with the same underlying network (IMP), both semantic distillation and internal knowledge distillation outperforms the network without distillation by a significant margin (resp. 6.8 and 8.6 points on RelCls). However, internal knowledge distillation outperforms semantic distillation, as the number of relations is much smaller, making the estimation of the relation probability given two objects more accurate.

Similarly, for VG-IMP (Table 4.3.4) and VG-RMatters (Table 4.3.5), we notice small differences between the baseline and SK for ImMacro, which we attribute to the small number of relations and fewer similar classes. We also note that we were not able to obtain competitive results with IK+ on both sets, keeping a similar implementation, which is especially surprising on VG-IMP. We have not found convincing explanations for this result and leave it for further research. However, SK improves ClsMacro, especially on VG-RMatters. This is due to the increased entropy of the corrected distribution used in the training loss defined in Equation 4.22, which results in an increased entropy of the output distribution. Since theImMacro scores change in a small amount, this suggests that the semantic constraints are consistent with the inter-dependencies between relations.

	SGCls ImMacro			RelCls ImMacro		
	R@20	R@50	R@100	R@20	R@50	R@100
Pixels2Graph [102]	-	35.7	38.4	-	**82.0**	86.4
IMP [158]	-	43.4	47.2	-	75.2	83.6
SGP [58]	-	**45.5**	**50.8**	-	80.8	88.2
MotifNet [166]	37.6	44.5	47.7	66.6	81.2	88.3
Baseline (Ours)	37.1	44.0	47.2	**66.7**	81.3	**88.7**
IK+	36.7	43.6	46.9	65.7	80.2	87.5
SK (Ours)	37.2	44.2	47.4	66.5	81.2	88.6

	RelCls ClsMacro		
	R@20	R@50	R@100
MotifNet [166]	15.7	27.5	37.9
Baseline (Ours)	**17.6**	30.2	41.3
IK+	15.3	27.6	39.0
SK (Ours)	17.2	**30.5**	**42.3**

Table 4.3.4 – Results with Semantic Distillation on VG-IMP. In the case of fewer relation classes, semantic distillation does not improve recall. This suggests that the resulting bias does not outweigh improved learning of rarer relations in the general case. However, ClsMacro recall is improved, which shows the pertinence of our approach for uncommon relations.

	SGCls ImMacro			RelCls ImMacro		
	R@10	R@20	R@100	R@10	R@20	R@100
MotifNet [166]	26.3	38.0	55.5	39.6	58.4	87.8
Baseline (Ours)	26.7	38.5	**56.3**	**40.3**	**58.9**	**88.3**
IK+	26.1	37.7	55.7	39.4	57.5	87.2
SK (Ours)	**26.9**	**38.8**	56.2	40.0	58.5	88.2

	RelCls ClsMacro		
	R@10	R@20	R@100
MotifNet [166]	11.8	19.8	46.6
Baseline (Ours)	12.6	21.1	47.8
IK+	11.4	19.1	47.2
SK (Ours)	**13.4**	**21.5**	**50.3**

Table 4.3.5 – Results with Semantic Distillation on VG-RMatters. Results on VG-IMP translate to VG-RMatters with increased ClsMacro and slightly deteriorated ImMacro.

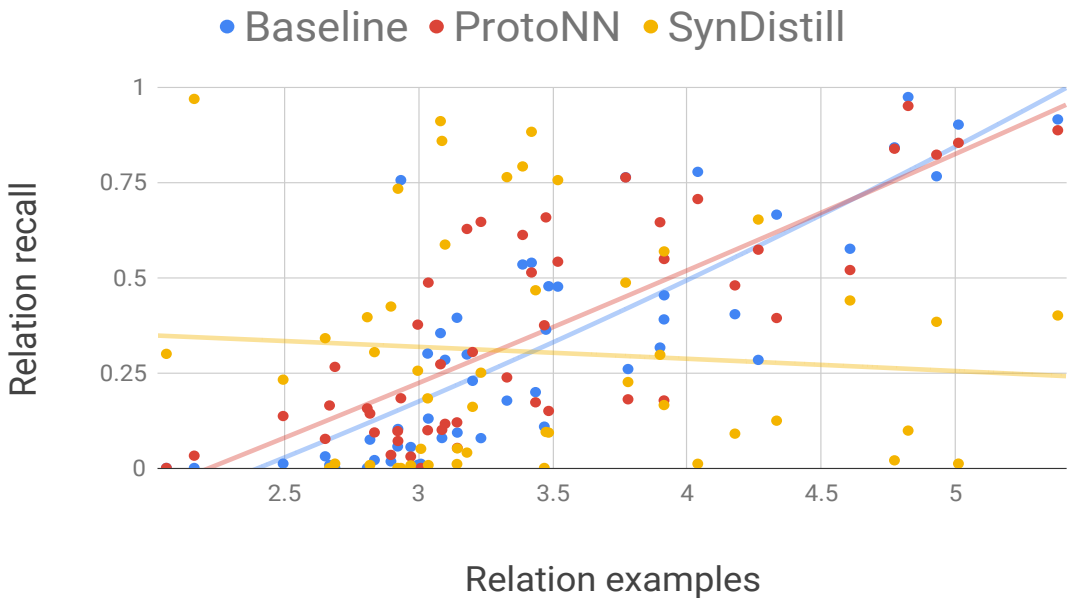


Figure 4.3.2 – Relation recall by the number of train examples in Visual Genome and trends for Baseline, ProtoNN and SynDistill. ProtoNN is able to detect with increased recall relations with fewer examples but is outperformed by our Baseline on the most frequent ones. SynDistill outperforms both approached for rarer relations, but its performance shows no correlation to the number of examples and performs especially poorly on the most frequent ones. This makes this model unreliable and suggests that it trains with a very inaccurate estimation of the loss function.

4.3.5.1 Result analysis

In Figure 4.3.2, for BASELINE, SYNDISTILL and PROTONN, we display the performance of the model by the number of training samples for each relation of VG. SYNAUGMENT is not displayed as the performance are very similar to if lower than CE. PROTONN has higher performance at a lower number of examples and is able to detect significantly more examples of the rarer classes than CE. Figure 4.3.3 shows the performance for the 30 most frequent relations in VG, sorted by decreasing frequency in the training set. We note that classes *under, at, against* and *watch* are significantly less detected by PROTONN. This is explained by a study of the prototypes, where clusters with these relations are populated with relation triplets with similar object pairs but have a low diversity of relations, which makes retrieving the less frequent ones less likely. Interestingly, SYNDISTILL has a much higher performance for several rarer relations. However, the performance on the most frequent relations are very low, which explains the difference in macro recall. We found no correlation between the number of synonyms of a relation and its recall. However, we notice that most relations with a high recall are actions whereas most actions with a much lower recall than the baseline are spatial actions.

Finally, Figure 4.3.4 displays the confusion matrix of SK and the difference between normalized confusion matrices of SK and our baseline. Positive numbers mean that the number of classified examples is greater in SK and negative for BASELINE. The class with the most positive changes is *riding*, which has the highest value at the diagonal , which is explained by Figure 4.3.5. This matrix shows the probability distribution for each relation, as defined in 4.27, from which we have set the diagonal to zero, as it is much higher than other values. The semantic similarity between *riding* and other relations is close to *zero*.
Two other relations have many more predicted examples: *near* and *holding*. For class *near*, this is not directly explained by the relation similarities, as many relations have more examples classified as *near*. This is also apparent by the fact that no more examples of relation *at* are classified as *looking at*, which is the relation with the highest probability given that *at* has been classified (due to the presence of *at* in *looking at*). This highlights one drawback of using similarity between word embeddings. However, it also shows that this approach is robust to noisy relationships as this similarity does not impact the performance on this class. Relation *holding*, however, is an example where the semantic similarity is useful, as many the similarity between relations *carrying* and *holding* results in more correctly predicted *holding*, as well as several examples of relation *carrying* classified as *holding*, which is implied by *carrying*.

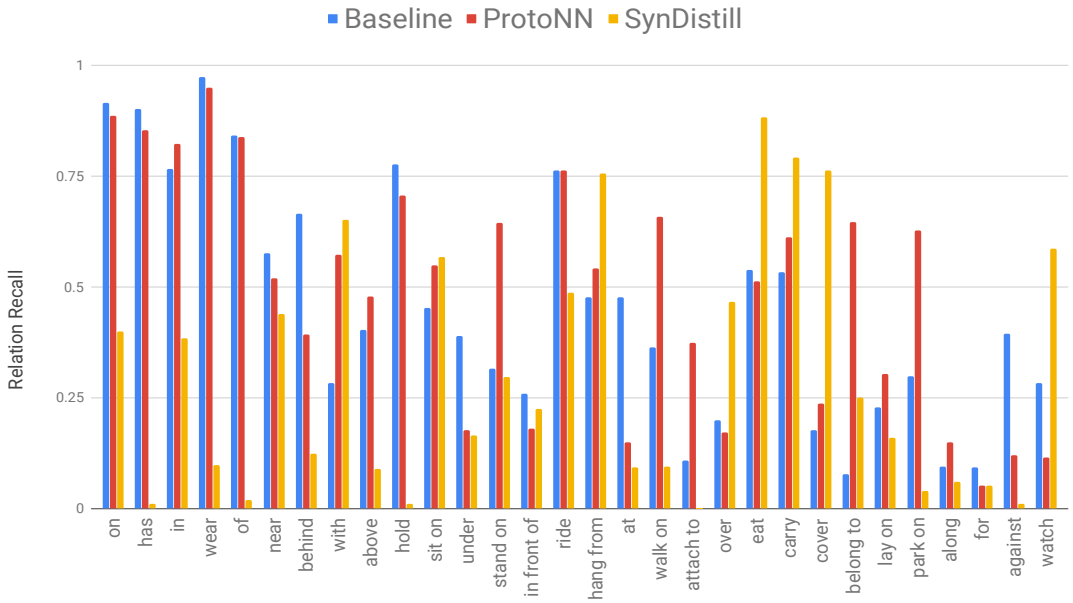


Figure 4.3.3 – Results of BASELINE, PROTONN and SYNDISTILL on Visual Genome for the 30 most frequent relations, sorted by decreasing frequency in the training set. PROTONN significantly increases recall for the rarest classes, by more than 100% for "*stand on*", "*park on*", "*belong to*" with slight decreases for the most frequent classes. However SYNDISTILL strongly deteriorates recall for the most frequent classes which negatively impact the overall recall.

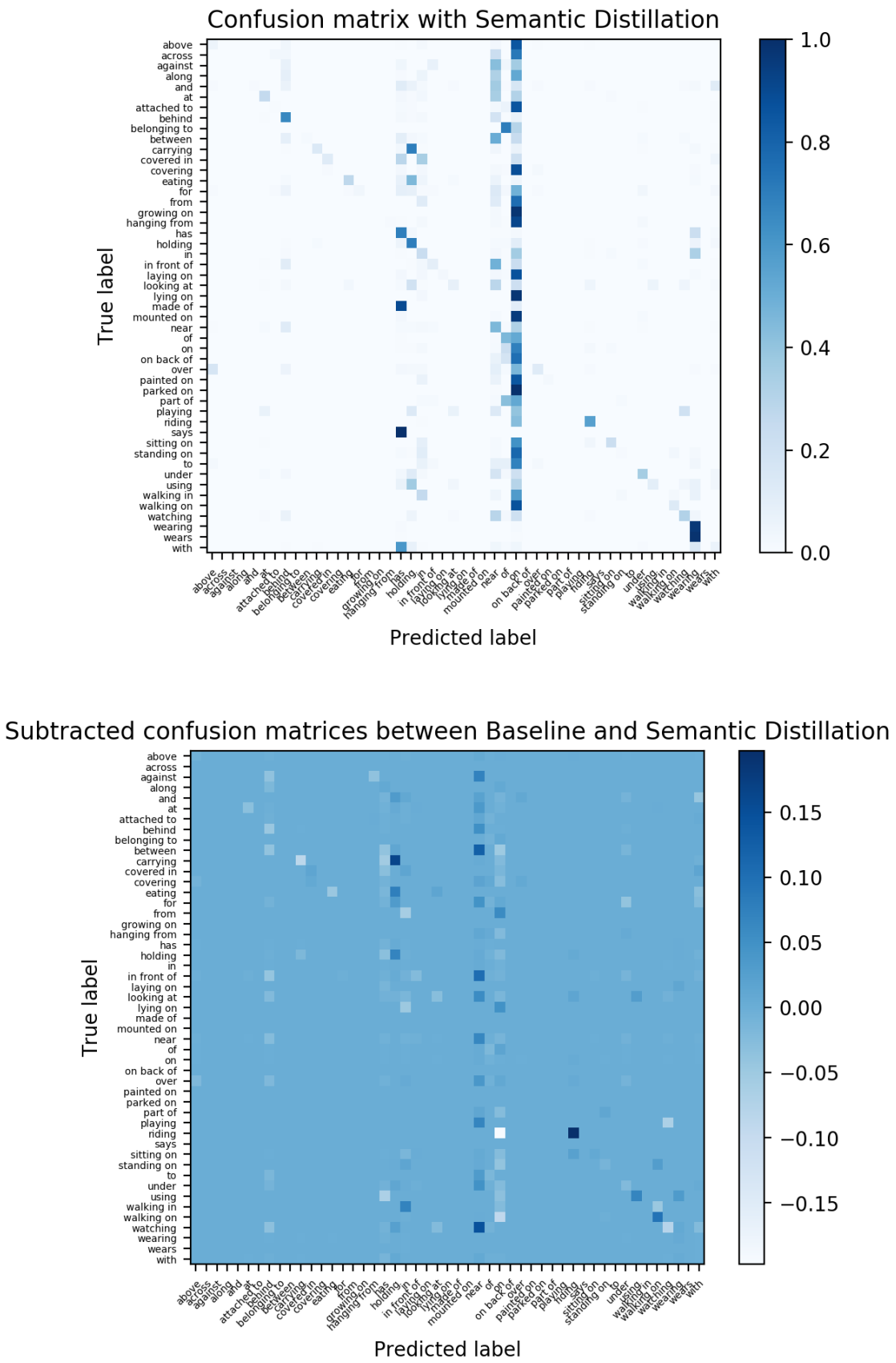


Figure 4.3.4 – Confusion matrix of SK (*top*) and difference with the Baseline confusion matrix (*bottom*).

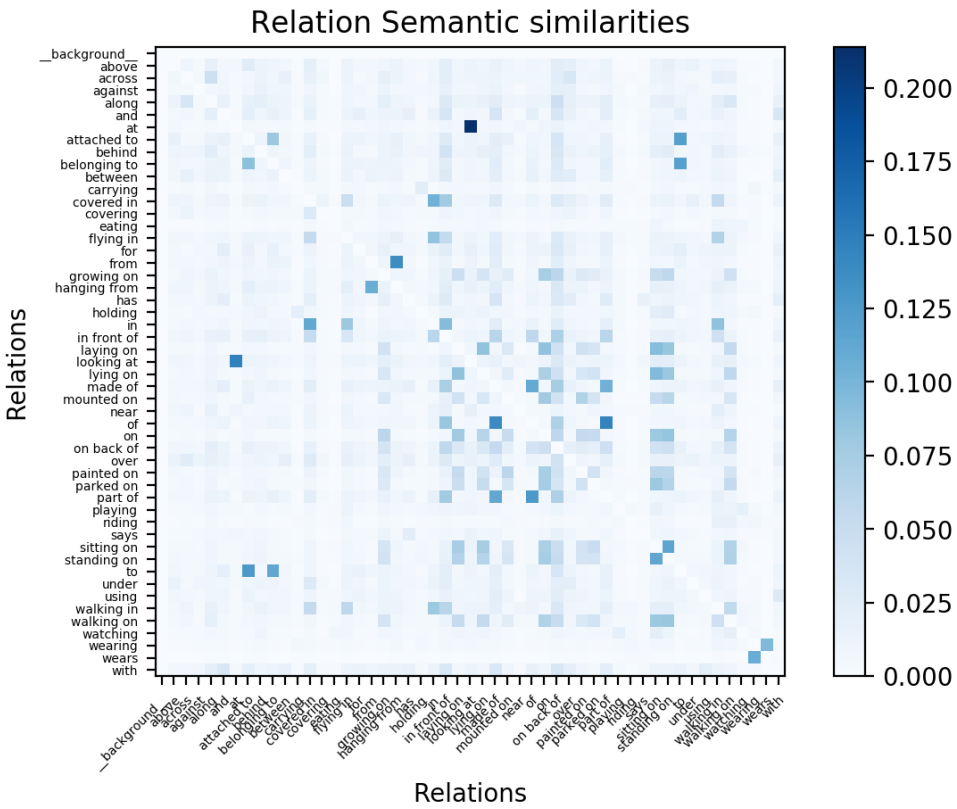


Figure 4.3.5 – Similarity matrix between relations in VG

4.3.6 Discussion

The results of models which use explicit synonymy modelling show that actions, which share more objects than spatial relations benefit more from comprehensive annotations. This suggests that existing models are capable of using correlations between object types and relations to correctly predict actions but struggle to predict spatial relations independently from object classes.

Furthermore, as shown by the prototype representations, due to the polysemy of relations, relations are synonymous only in a set of cases. Thus using a rigid dictionary independent of the involved objects introduces a bias in the relation distribution estimation.

Finally, distillation-based methods have proven less effective on datasets with fewer relations. Several hypotheses can explain these limitations. The first is the differences between semantic and visual similarities. With word embeddings, relations *behind* and *in front of* have high cosine similarity because of a high co-occurence rate but they are very visually different. Similarly, *standing on* and *crossing* are usually true at the same time, but they are not synonyms. This makes less robust methods such as data augmentation not adequate for the task. Rule distillation proves more useful and robust in this case, as the added loss term depends on the output distribution and thus, relationships between classes which do not translate to the visual domain have less impact on the trained model.

4.4 Conclusion

We studied three methods defined with the purpose of removing the assumption that relations are exclusive. The first by defining a metric space in which relations are embedded. This allowed us to study the defined relation representations and highlight a trade-off between using only visual data and adding external knowledge in the form of object similarities, which helps the model learn common sense, predicting common relations for given object pairs. However, this also biases the model toward common relations and makes it harder to determine which features can really separate relations with high intra-class diversity, such as *on* and *near*. Results show that prototype learning is promising, when it comes to predicting a larger variety of relations while maintaining a high average recall. The comparison with two methods of explicit modelling of synonyms highlights that implicit modelling of synonyms (with prototypes) has the advantage of being adapted to the visual domain, where visual relations have many different meanings and are not synonym to the same relations as text relations.

Finally we showed that it is possible to use external knowledge in order to improve

the recall of rare relations. Rule distillation based on semantic similarities between relations helps improve the recall of uncommon relations. This is especially useful where relations are less restricted, as is the case for VG-Large. Indeed, in VG-large relations can be comprised of several words and thus are related to many other classes. We showed that semantic and internal distillation obtain similar results on both VG-Large and VRD-set, with the caveat that internal distillation requires additional pre-processing for large numbers of classes and that is better adapted to datasets with more restricted classes. This approach also provided very good results on VG-RMatters, improving the macro recall@10 from 12.6% to 13.4%, with a slightly lower micro recall however. This shows that this approach is relevant in the case of a more balanced dataset, improving the estimated relation distribution and the recall of several relations, and decreasing the bias toward the most common ones.

Chapter 5

Relation Relevance

5.1 Context

5.1.1 Motivation

Existing methods are able to retrieve a high number of relations, However, because they are trained on annotations from existing datasets, they are heavily penalized when these relations are missed and will favor them above rarer ones. As shown in Section 3, existing methods have significant biases. This results in a very low recall for the latter relations. In Section 4, we tackled the imbalance between relation classes, aiming to improve the representation of rarer classes by using the relationships between rare and common classes and learning from the latter, which provides many more examples to learn from.

However, the observation of a few images of Visual Genome showed that existing models tend to predict one relation per object pair. Furthermore, the study of the confusion matrix with the background class shows that the model is not able to detect whether a relation should be annotated: the background class (the leftmost class) has nearly always a higher probability than other relations.

This makes Visual Relation Detection different from object detection, because the relevance of a relation depends on a large variety of criteria. First of all, it depends on characteristics of the objects, such as their size, saliency, distance. Second, it depends on the nature of the relation: if it is too typical, then the relation might not be mentioned, unless the objects are central in the image, the image has a low number of objects... In the case of Visual Genome; the relevance of a relation is harder to estimate, because it depends on whether a region of the image has been described by a previous annotator, as shown in Figure 3.1.2. Finally, relation annotations depend on the knowledge of the annotator, their previous experiences and so on... Figure 5.1.2 illustrates this, where the relation (*tree, has, bark*) is true in both images but is not annotated in the rightmost image.

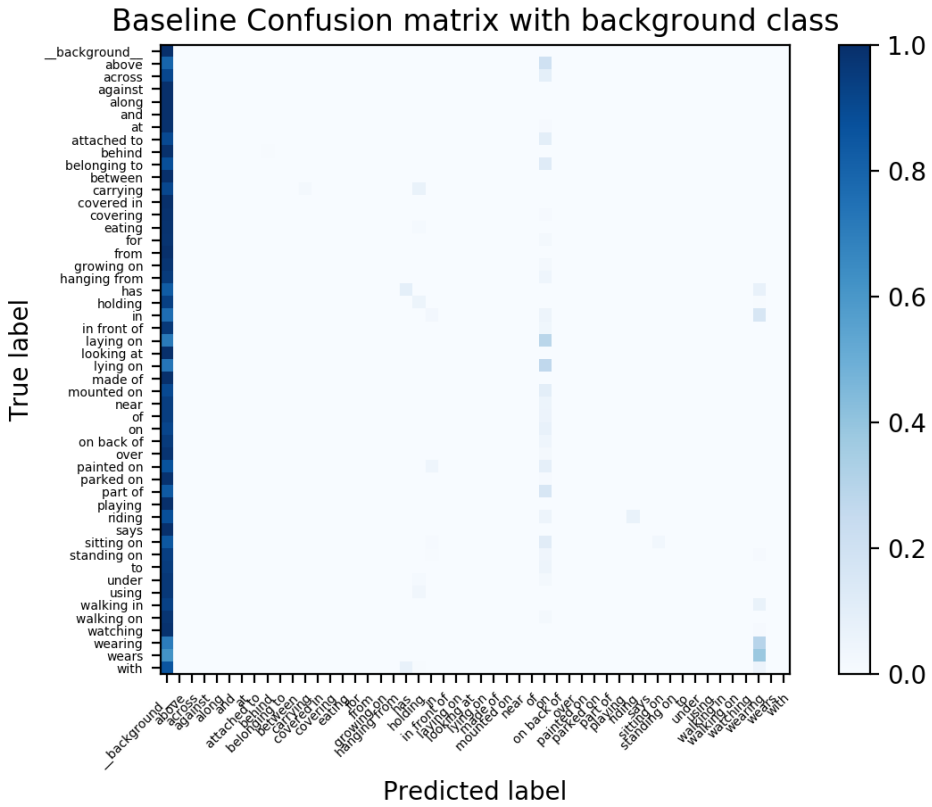


Figure 5.1.1 – Confusion matrix of our Baseline on VG-RMatters

Figure 5.1.2 – Examples from Visual Genome. The relation *(tree, has, bark)* is annotated in *leftmost* image but not in the *rightmost* image.

For humans, it is important to be able to filter out sensory information, as people who are not able to do so have trouble processing all the available information. Thus some parts of image may be ignored without conscious decisions. We argue that a model able to better detect relevant object pairs and relations will result in a higher diversity of predicted relations. Furthermore, it will predict more relations relevant to a human observer and also learn to filter out information that a human would ignore. We propose a method to train a relevance estimator between objects. This method directly targets one of the obstacles discovered in Chapter 3, namely the fact that the objective function and target metric do not correlate. While Chapter 4 targeted the evaluation of the experimental risk, related to the objective function, this Chapter is dedicated to improving the inference process.

In this Chapter, we aim to improve performance on relation classification by prioritizing object pairs that are the most relevant, i.e. which are the most likely to be annotated by a human annotator and we show two complementary ways to achieve this goal.

Contributions Our contributions are summarized as follows:

1. *Latent topic modelling of visual relations.* To exploit the dependencies between annotated relations and image content, we propose to use relations as text documents to extract high-level information in the form of latent topics with Latent Dirichlet Allocation [10]. We show that adding a topic prediction layer improves the relevance of predicted relations. We use an attention mechanism to aggregate relation features in order to predict image topics and to use this attention as a factor in the relevance score.
2. *Relevance classifier with Prior Potentials* We propose a simple new method to generate scene graphs with guided proposals using a relevance score and statistics-based priors, aiming to increase the diversity of predicted relations.

5.1.2 Related Work

Concept Relevance refers to the phenomenon whereby the probability of that a concept is visible is different from the probability that it is annotated by a human. Berg et al. [9] showed that objects either far from the center of an image or small are less likely to be mentioned. Unsual objects and people however, tend to be mentioned more often. Misra et al. [99] tackle this discrepancy by separately modelling the presence of an object and its relevance so that the model may simultaneously predict a high probability of presence and a low relevance.

The selection of the relationship proposals was studied by Zhang *et al.* [168] where they point out the difficulty of choosing relation proposals (i.e. object pairs

to be annotated) when the number of object proposals grows, as the number of possible relations grows quadratically. They learn a relation proposal network based on visual and spatial features and they evaluate this network on Visual Genome [71] and show that the visual compatibility score is more important than the spatial compatibility score.

Finally, Zellers *et al.* [166] show that, in contexts where ground truth object detections are not given, a baseline predicting the most frequent relation outperforms existing methods such as [89, 102, 158] by filtering out object pairs that do not overlap.

5.1.3 Formulation

Both contributions rely on the following decomposition of scene graph probability:

$$P(G) = \prod_i P(V_i) \prod_{j \neq i} P(R_{i \to j}, Z_{i \to j} | V_i, V_j) \tag{5.1}$$

This formulation takes into account the presence of a relation with $R_{i \to j}$ and its relevance to a human observer with $Z_{i \to j}$. Taking into account the observation that many true relations should not be mentioned, this makes the prediction of the model focus on a smaller number of object pairs and extract more relevant relations from an image. Here, we make the assumption that they are independent variables. In this Chapter, we describe how the relation distribution $P_{vis}(R_{i \to j} | V_i, V_j)$ and human relevance distribution, $P_{\text{human}}(Z_{i \to j} | V_i, V_j)$ are computed.

5.2 TopicNet: Learning Relation Representations with Attention to Topic

We present our model for extracting scene graphs from images using Topic Modelling and Attention to relations, called TopicNet.

5.2.1 Motivation

In Chapter 3, we showed that the model struggles to reliably retrieve all relevant classes. To improve the pair selection process, we propose to take inspiration from other self-supervized methods described in Section 2.4.2. Indeed, as previously mentioned, human observers tend to not mention objects that are typical, too small and so on. From this observation, we make the hypothesis that relations are considered relevant if they involve objects that are relevant to the image topic. We call topic latent variables that condition the probability of visible objects. Thus if a

given relation helps to predict the image topic, then it involves relevant objects and therefore is considered as relevant.

5.2.2 Related Work

Relations are annotated not in isolation, but are all drawn from a distribution conditioned by the image they describe, which not only conditions which relations are true, as has been noted in [158], but also their relevance. Hence, we assume that they are drawn from a distribution conditioned by the same latent variable, which we call the topic of the image. This topic is estimated by Latent Dirichlet Allocation, by Blei, Ng and Jordan [10], where the documents are the set of relations of each image. Furthermore, we show that the relation between each object pair and the image topic carries a significant role in predicting how important the pair is for understanding the image content and thus how likely it is to be relevant to a human. For this, we train a network to predict the image topics using an attention mechanism and show that, at test time, this attention prediction can be used as an indicator of object pair relevance.

Image context is highly influential on whether an object will be mentioned by a human viewer, as was shown by Berg et al. [9]. We aim to show that this is also true for the relations that a human viewer chooses to focus on, as this information is critical for the estimation of the relevance of a relation in the given image. Furthermore, as the number of possible pairs to annotate is quadratic with respect to the number of objects, it is vital to focus on important object pairs. For Visual Relation Detection, this information (in the form of scene labels for example) is usually not directly available. However, with the available relation annotations, we have for each image a set of *(head, relation, tail)* triplets. In the same vein as Gomez et al. [52], we use the Latent Dirichlet Allocation (LDA) by Blei et al. [10] to model topics of images using relation phrases. Gomez et al. [52] show that the added self-supervision from LDA topics enables the network to learn visual features that provide performance comparable to supervised methods on downstream tasks, such as object classification and multi-modal image retrieval.

5.2.3 Describing images with latent topics

Formulation Given a vocabulary of words $\boldsymbol{w} = (w_1, \ldots, w_V)$ and topics $\boldsymbol{z} = (t_1, \ldots, t_K)$ and a corpus D consisting of M documents each of length N_i, LDA with parameters α and β assumes the following generative process for document d_i

— Choose $\theta_i \sim Dir(\alpha)$

— Choose $\phi \sim Dir(\beta)$

— For each position k in d_i:

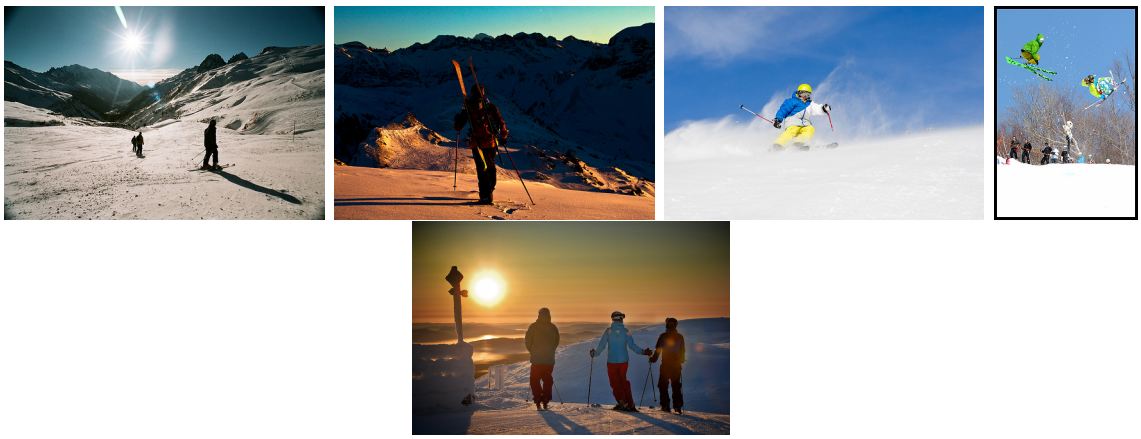
— Choose a topic $t \sim Multinomial(\theta_i)$

— Choose a word $w \sim Multinomial(\phi_z)$

After inference, two distributions $P(\boldsymbol{w}|\boldsymbol{t})$ and $P(\boldsymbol{t}|\boldsymbol{d})$ are available.

Pair annotations as documents An object being in a different role than usual might be an indicator of a different image topic. Thus, to train the LDA, we use the object pairs of the image relations as the sentences of the image document. For instance, the relational phrase (rider, on, horse) becomes '*rider horse*'. Once the documents have been defined, we first train the LDA and use the detected topics as topic ground truth during the training of the neural network.

Figure 5.2.1 displays, for three different topics, the images with highest probability of each topic to be drawn for the image. Topics describe classes of objects that interact frequently and are highly interpretable. A correct topic prediction is an indicator for the ability to determine relevant object pairs in the image.



(a) Subjects: boot, coat, glove, hat, helmet, jacket, mountain, pant, ski, snow. Objects: person, ski, skier, snow, track.

(b) Subjects: bike, cat, people. Objects: bike, cat, paw, people.

(c) Subjects: arm, banana, hand, head, logo, number, player. Objects: banana, cap, glove, hat, helmet, logo, number, player, sock.

Figure 5.2.1 – Top associated objects and images of two topics inferred with Latent Dirichlet Annotation. Topics are consistent, although we note that some contain semantically contrasting words, such as ‘cat‘ and ‘bike‘, which will increase the likelihood that relations between the two will be predicted.

5.2.4 Visual Relation Detection with Attention to topic

We present the steps taken from the extraction of the representation of relational phrases to the prediction of topic and the phrase itself. The process is described in Figure 5.2.2. In the following section, we will refer to learned weights and biases as W^* and b^*.

Topic prediction with Attention Our goal here is to show that image topics can be inferred from relation representations. For that, we use an attention model relying on the image representation, as shown in Figure 5.2.2, inspired by recent works on visual question answering [2] that rely on a query to determine which object detections are important to finding the answer. The underlying intuition is that the image context itself can be used as a query to determine which relation is important for the topic classification. The model learns which object pair and image features have a high correlation and how they correlate with the different topics.

For each object pair (h, t), we define:

$$q_{h,t} = \left(W^{rel} \cdot \boldsymbol{f}_{h,t} + b^{rel}\right) \odot \left(W^{im} \cdot \boldsymbol{f}_I^{vis} + b^{im}\right) \tag{5.2}$$

where $\boldsymbol{f}^{vis}$ corresponds to the image features extracted by the CNN and $\odot$ is the element-wise product operation. Thus $q_{h,t}$ is the query defined from the relation and image features, from which the attention to the relation is defined as follows. The attention weight $a_{h,t}$ is defined for each object pair (h, t) from the query $q_{(h,t)}$:

$$\alpha_{h,t} = W^{att} \cdot q_{h,t} + b^{att} \tag{5.3}$$

$$a_{h,t} = \frac{e^{\alpha_{h,t}}}{\sum_{k,l \in [1,N_I]} e^{\alpha_{k,l}}} \tag{5.4}$$

Finally, the image topic embedding is defined as a weighted sum of each object pair representation:

$$\tilde{x}_I^{topic} = \sum_{h,t \in [1,N]} a_{h,t} q_{(h,t)} \tag{5.5}$$

The topic distribution on I, noted $p_\Theta(T|I)$, is predicted using a fully connected layer followed by softmax operation on $\tilde{x}_I^{topic}$ as shown in Figure 5.2.2. weights are learnt by minimizing a binary cross-entropy loss between LDA topics and predicted distribution:

$$\mathcal{L}_{topic}(\Theta, I, p_{lda}) = \frac{1}{K} \sum_{t \in T} p_{lda}(t) \log p_\Theta(t) + \left(1 - p_{lda}(t)\right) \log(1 - p_\Theta(t)\big) \tag{5.6}$$

For conciseness considerations, we abbreviate $p(T = t|I)$ by $p(t)$.

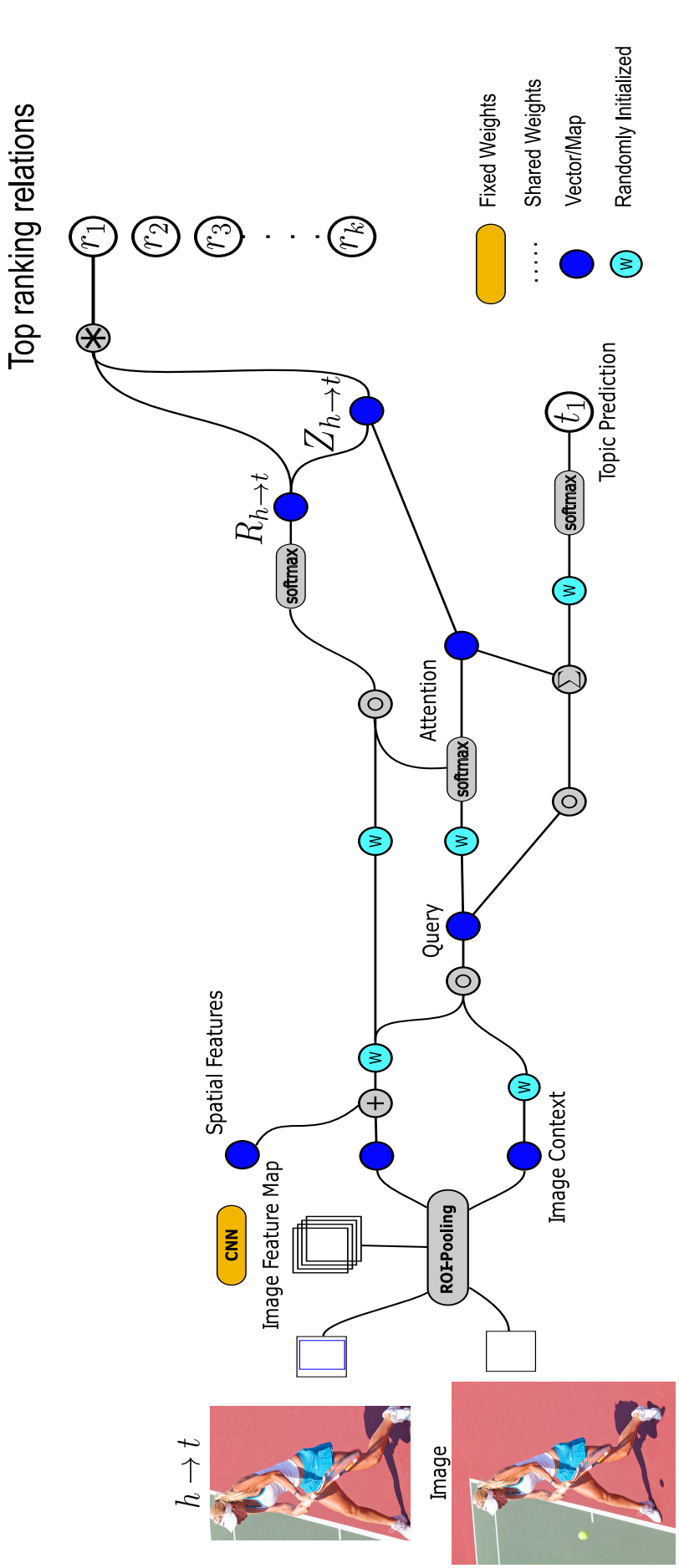


Figure 5.2.2 – TopicNet framework. On the upper branch, relation classes are predicted from concatenated image an relation features to incorporate image context into the prediction. On the lower branch, image and relation features are combined to define a topic query. Attention over each object pair is computed from the query vector to aggregate a global context and predict the latent image topics. At test time, the predicted attention is used as a relevance indicator for scene graph generation.

Relation prediction from object pair representation To predict the relations of object pair (o_h, o_t), the same relation embedding is passed into a feed-forward layer and added to the spatial and object logits, as for our Baseline shown in Figure 3.3.1. Then a softmax operation outputs the probability distribution over the set of relation classes $\mathcal{R}$. This branch is trained with a cross-entropy loss $\mathcal{L}_{rel}$ between the ground truth distribution and the output distribution.

At training time, the following weighted loss is optimized

$$\mathcal{L} = \alpha\mathcal{L}_{rel} + \beta\mathcal{L}_{obj} + \gamma\mathcal{L}_{topic} \tag{5.7}$$

where α, β, γ are determined experimentally.

Relevance prediction from object pair representation Following Section 5.2, we have extracted attention weights over each visual relation, which indicate how much a relation correlates to the image topic distribution. Thus, we model this relevance as the attention that the model estimates to predict image topic:

$$P(Z_{h,t}) = \log a_{h,t} \tag{5.8}$$

5.2.5 Experiments

We evaluate the performance of our model for relation classification on the VG-IMP [158] and VG-RMatters splits.

Implementation details We use the loss defined in Equation 5.7 with $\alpha = \beta = 1, \gamma = 5$ selected among values $1, 5, 10, 20, 30$. All TopicNet variations are trained with 90 topics, chosen experimentally (performance is stable throughout the tested configurations, from 10 to 100 topics).

We study the impact of TopicNet with two different underlying networks. The first is comprised of one stream for relation classification, the more usual VRD model, as shown in Figure 5.2.2. The second corresponds to our baseline, with four streams, as in Figure 3.3.1. All weights are initialized using normal Xavier intialization [50].

Comparative results In Table 5.2.2, we compare our proposed method to state of the art approaches. Pixel2Graph [102] iteratively refines object and relationship heatmaps with a stacked hourglass network making use of global context. IMP [158] iteratively learns to refine relationship and object representation by passing mesages through the scene graph. MotifNet [166] captures higher order correlations between objects and relationships using LSTM layers. SGP [58] is a permutation invariant graph predictor that refines predictions from MotifNet using attention

over linguistic and visual neighbor features. AR (Attention Relevance) is the relevance factor defined in Section 5.2.4.

For one and four streams, we notice the same trend: TopicNet slightly deteriorates performance on image macro recall. Furthermore, attention relevance is even more detrimental for this metric. However, with one stream, class macro recall is signficantly improved by both TopicNet and AR, from 36% to 41%. This does not translate to four streams, however. On VG-RMatters, results have small variations between the three configurations with no identifiable trend, as shown in Table 5.2.1. Tests with one stream were done without the change to the early stopping presented in chapter 3, thus this suggests that the improvements brought in this case may not be statistically significant. New tests with 1 stream could help confirm this hypothesis. However, we conclude here that this approach to the estimation of relevance is not conclusive.

	SgCls		RelCls			
	ImMacro		ImMacro		ClsMacro	
	R@10	R@20	R@10	R@20	R@10	R@20
MotifNet [166]	26.3	38.0	39.6	58.4	11.8	19.8
Ours - 4 streams						
Baseline	26.7	38.5	40.3	**58.9**	**12.6**	**21.1**
TopicNet	**27.1**	**38.8**	40.0	58.5	12.3	20.1
TopicNet - AR	**27.1**	**38.8**	**40.8**	58.7	12.4	20.3

Table 5.2.1 – Results with 4 streams on VG-RMatters. Recalls are in % and evaluated without scene graph constraints. TopicNet provides slight improvements over our baseline on SgCls and RelCls ImMacro but results are inconclusive.

	SgCls			RelCls		
	ImMacro			ImMacro		
	R@20	R@50	R@100	R@20	R@50	R@100
Pixel2Graph [102]	-	35.7	38.4	-	**82.0**	86.4
IMP [158]	-	43.4	47.2	-	75.2	83.6
SGP [58]	-	**45.5**	**50.8**	-	80.8	88.2
MotifNet-NoCtxt [166]	36.4	43.1	46.3	64.6	78.8	86
MotifNet [166]	**37.6**	44.5	47.7	**66.6**	81.2	**88.3**
Ours						
1 stream						
Baseline	**36.9**	**43.7**	46.9	**64.9**	**79.4**	86.7
TopicNet	36.6	43.6	**47.1**	64.7	**79.4**	**86.9**
TopicNet - AR	34.9	42.5	46.2	60.5	75.8	84.4
4 streams						
Baseline	**37.1**	**44.1**	**47.2**	**66.7**	**81.3**	**88.7**
TopicNet	36.9	43.8	47.1	66.3	81.0	88.3
TopicNet - AR	35.7	43.1	46.5	64.0	79.5	87.3

	RelCls		
	ClsMacro		
	R@20	R@50	R@100
MotifNet [166]	15.7	27.5	37.9
Ours			
1 stream			
Baseline	15.7	26.6	36.0
TopicNet	17.1	29.2	40.0
TopicNet - AR	**18.0**	**30.3**	**41.0**
4 streams			
Baseline	**17.6**	**30.2**	**41.3**
TopicNet	16.1	28.8	39.9
TopicNet - AR	15.4	27.7	38.5

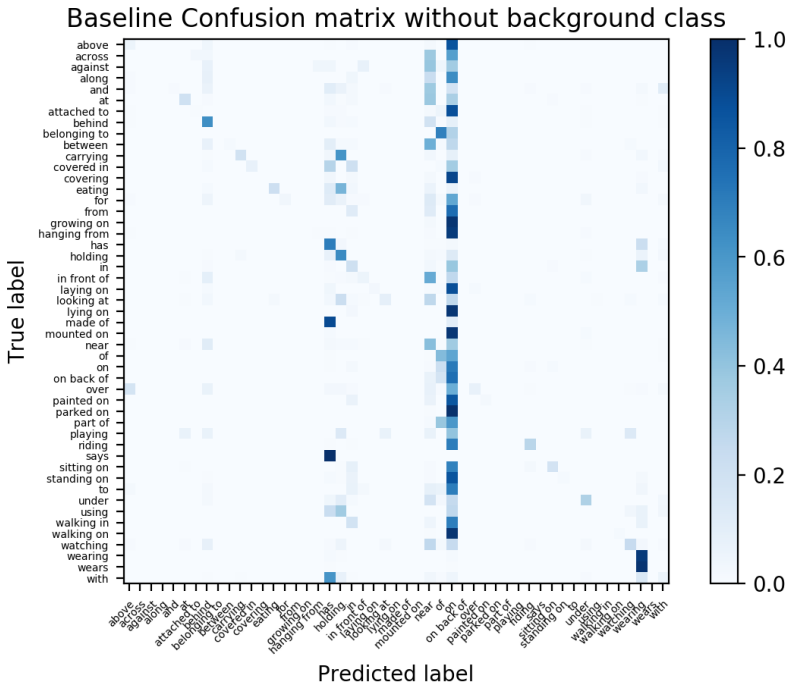
Table 5.2.2 – Results of TopicNet with 1 and 4 streams, tested on VG-IMP. Recalls are in % and evaluated without scene graph constraints. TopicNet is outperformed by our Baseline in the 4-stream. Estimation of relevance with topics is not conclusive.

5.2.6 Discussion

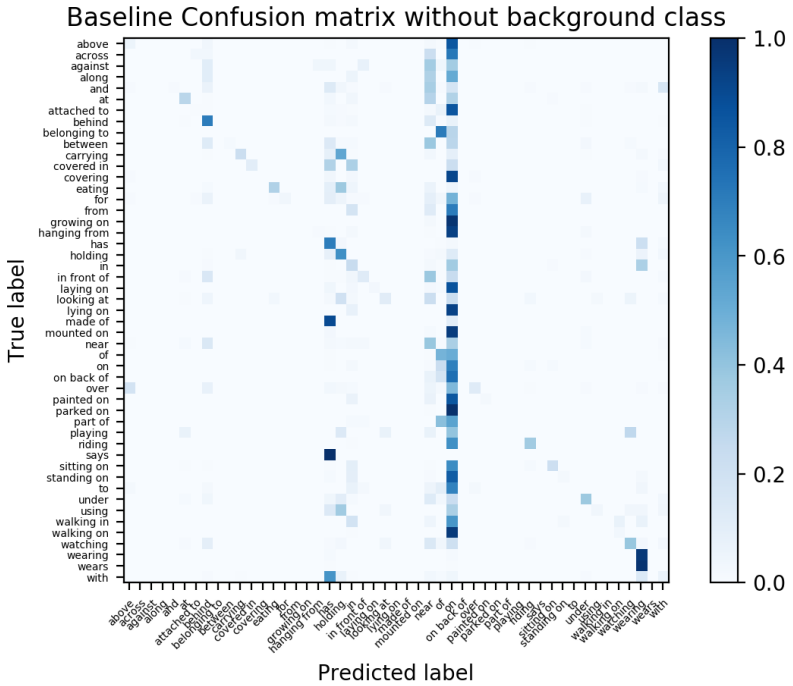
From the previous observations, we make two conclusions:

— Improvement in classification brought by the 4 streams makes it less necessary to predict several relations per pair. We verify this by comparing the confusion matrices between 1 and 4 streams in Figure 5.2.3. Classification of the following classes is improved (as shown by the increased proportion in the diagonals): *at, behind, eating, covered in, holding, in front of, of, riding, under, watching.* This is one explanation for the reduced improvement brought by TopicNet-AR, which is aimed at increasing the performance of rarer relations.

— Either the attention to each relation for the image topic is not correlated to its relevance or the network is unable to correctly learn which relation to focus on. To verify this, we tested the results on topic classification. For this, measure the recall when predicting the top k scoring topics, shown in Figure 5.2.4. Since several topics are associated to similar objects, the top 1 recall does not completely reflect the performance on topic classification. At 6 predictions, the recall reaches 91%. We conclude that the network learns to classify the image topic. Hence, this suggests that relevance and predicted attention are not correlated. Further study of images among the different topics distribution would be helpful in confirming or invalidating this argument.



(a) 1 stream



(b) 4 streams

Figure 5.2.3 – Confusion matrices for relation classification with one stream (*top*) and four streams (*bottom*). Increased performance between 1 and 4 streams for *at, behind, eating, covered in, holding, in front of, of, riding, under, watching* is a first explanation for reduced performance of TopicNet.

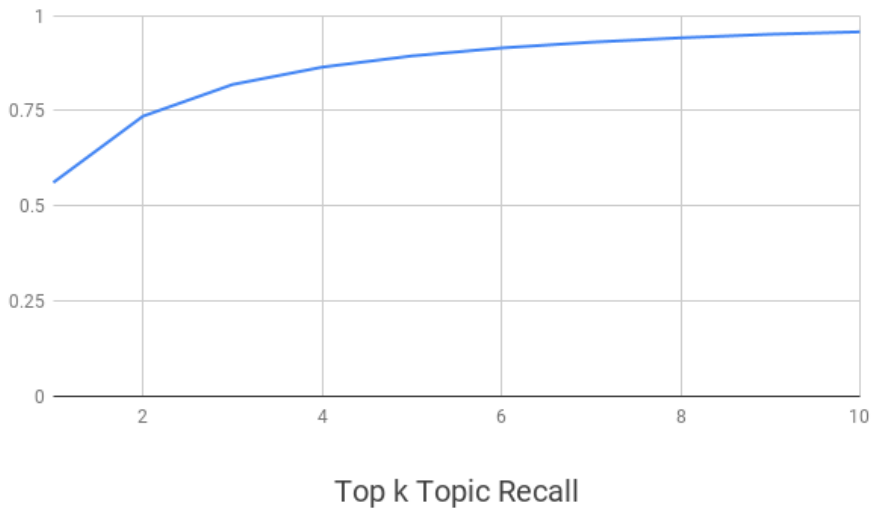


Figure 5.2.4 – Top k recall for topic classification among 90 topics. Due to the redundancy between topics, we focus on recall after several predictions. At 6 predictions, the recall is at 91%. We conclude that the network learns to classify the image topic.

5.2.7 Conclusion

To sum up, we presented a novel relation detection method based on topics extracted from images according to the visual relations that occur in them. A deep neural network is trained to predict topics in unseen images by aggregating context from each visual relation. For that, an attention mechanism weighs the contribution of each relation to the context and thus to the topic of the image. As the network learns to pay attention to specific relations in the image, it learns to determine which features correlate the most to the image topics

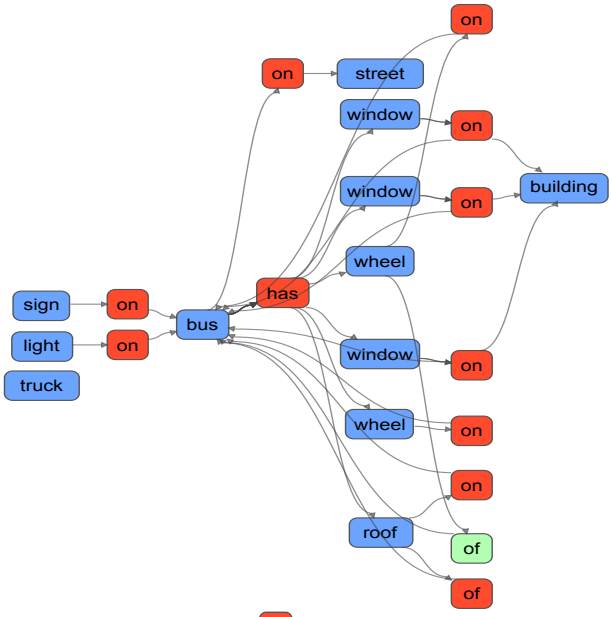
We have proven that the network is able to predict the topic of the image. However, when the relation classification reaches a threshold of accuracy, integrating the estimated relevance of each object class does not provide any improvement to the global recall. We concluded that the estimated attention, used in order to predict the image topic, is not correlated to the relevance of the object pair.

Thus, in the next section, we explore how to model the relevance and improve results on the 4-stream network.

Most probable words from detected topics:

— SUBJECTS: *building, bus, street*

— OBJECTS: *door, windshield, car, wheel, bus, window, tire*



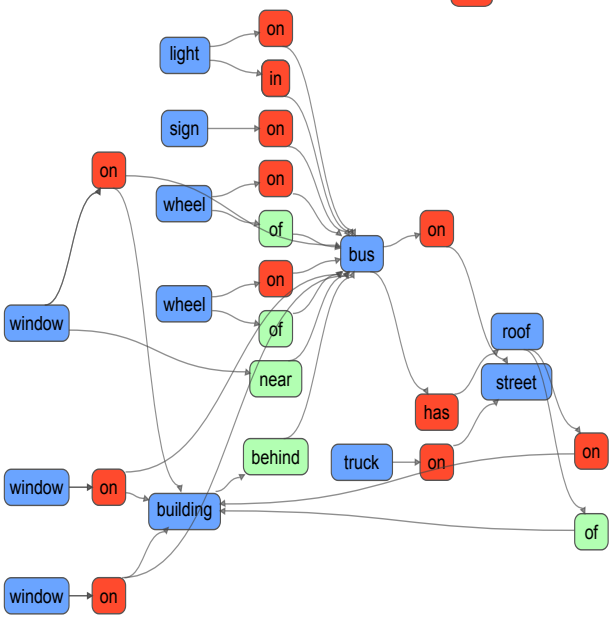


Figure 5.2.5 – Extracted scene graphs for MotifNet (top) and TopicNet (bottom). True relation predictions are *green* and false positives *red*. TopicNet is able to identify some relations to focus on using relevance prediction.

5.3 Focused VRD with Prior Potentials

5.3.1 Motivation

We previously showed that adding a relevance factor helps improve the class macro recall on a less performing classifier. However, this relevance failed to provide improvements on a better performing one. In this Section, we explore a different approach, aiming to train a relevance classifier. However, in existing datasets, a high number of relations are true in each image, and because of the previously described phenomena, only a small fraction of them are annotated. This makes it hard for supervised models to extract relation representations and boundaries that separate relevant and irrelevant relations.

5.3.2 Relevance Estimation with Prior Potentials

In the same vein as the semantic bias, introduced by Zellers *et al.* [166] and defined in Section 3.3.1, we model the relation relevance using two terms: a relevance classifier and a prior potential.

$$P_{\text{human}}(Z_{h\to t}|v_h, v_t) = p_\theta(Z|v_h, v_t) + \phi(v_h, v_t) \tag{5.9}$$

where $p_\theta(Z_{h\to t}|v_h, v_t)$ is a trained **relevance classifier** and ϕ is a **binary prior potential**. Thus, as shown in Figure 5.3.1 (d), the relation is estimated using both relation content as well as dataset statistics.

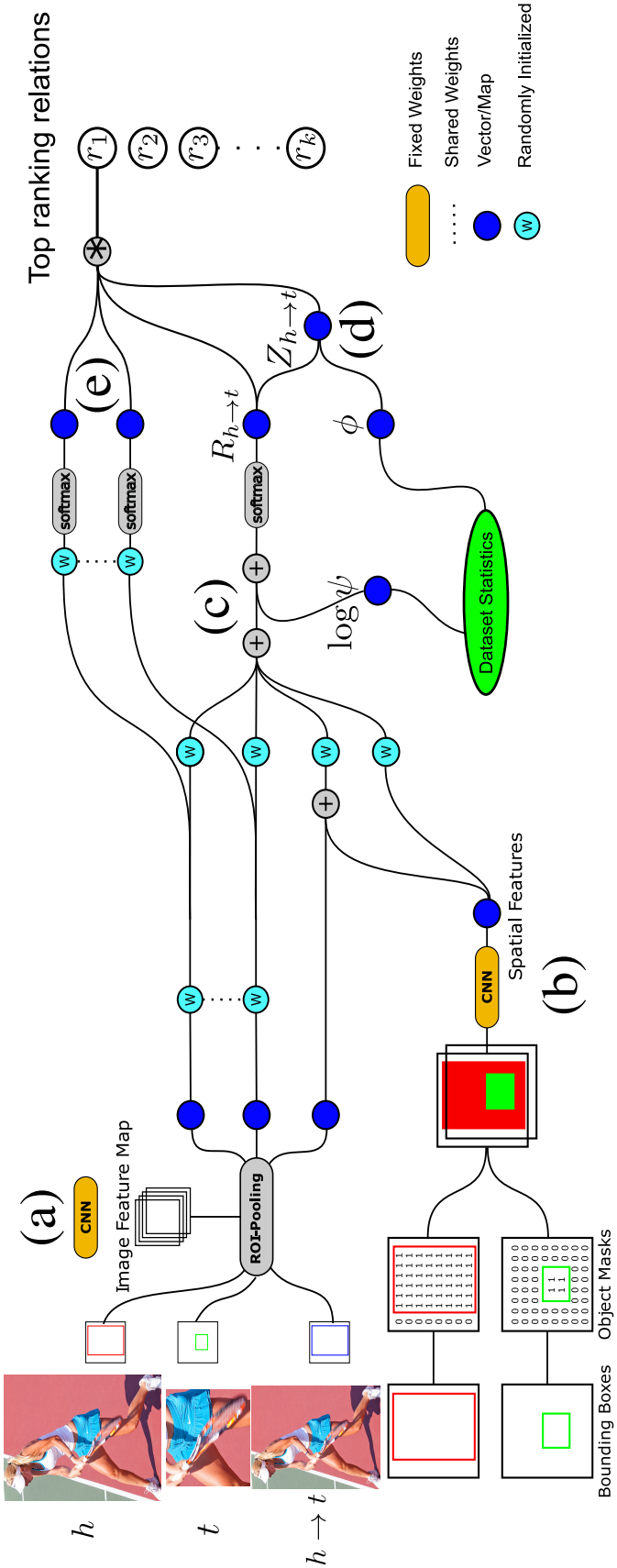


Figure 5.3.1 – FocusedVRD framework. Visual and spatial representations of each object pair is extracted using CNNs. Statistics-based priors are computed on the training set to predict relations more compatible with common sense (ψ) and improve relevance estimation (ϕ). At test time, a Scene Graph is generated by combining object classification probabilities (e), relation classification probabilities (c) and relevance probabilities (d). This allows the model to focus on relevant object pairs in the image and predict uncommon relations.

Relevance Classifier We model the relevance classification as the probability that any relation is annotated on the given object pair:

$$p_\theta(Z|v_i, v_j) = 1 - p_\theta(R_{i\to j} = \varnothing) \tag{5.10}$$

The underlying intuition is that most overlapping pairs of objects are related, with at least one spatial relation being true. Thus if the pair has not be annotated, then the true relations are not relevant. Thus, taking the contrapositive, the probability that these relations are relevant is greater than the probability that at least one is annotated, which corresponds to the probability in Equation 5.10.

Prior Potential of Relevance The relation potentials ψ and ϕ are inspired by [166, 170] where authors use a semantic module defined as the empirical distribution of relations given two objects, noting that the number of probable interactions between two object is limited. Similarly, the likelihood that two objects will share a relevant relation can be estimated by the frequency at which they interact in the training set.

This potential is used in addition to the relevance classifier because the relevance of relations in the training set has a high level of noise which makes the training of this classifier very unstable. This risks biasing the model predictions but makes use of identifiable trends in relation relevance which improves the relevance estimation.

The binary potential ϕ is computed by counting the number of co-occurences of both objects and the number of times they are in a share relation.

$$\phi(v_h, v_t) = 1 - P_{data}(R_{i\to j} = \varnothing) \tag{5.11}$$

$$= \frac{\sum_I \sum_{r\in\mathcal{R}} \mathbb{1}(v_h, r, v_t)}{\sum_I \mathbb{1}(v_h, v_t)} \tag{5.12}$$

We note that the use of unary potentials, where the frequency of relations are measured for each object separately does not provide any improvements, which suggests that this distribution adds too much noise. For example the pair *(man, horse)* has a much more limited range of possible relations than *man* alone, which, in the studied data, is paired more often with pieces of clothing.

5.3.3 Experiments

Comparative results In Tables 5.3.1, we compare our method to state of the art approaches on the VG-IMP split [158]. PIXEL2GRAPH [102] iteratively refines object and relationship heatmaps with a stacked hourglass network making use of global context. IMP [158] refines relation and object representation by passing mesages through the scene graph. MOTIFNET [166] captures higher order correlations between objects and relationships using LSTM layers. SGP [58] is a permutation invariant graph predictor that refines predictions from MOTIFNET using attention over linguistic and visual features of neighbors. RELDN [170] is not included because of different evaluation protocols, as object pairs without relations are filtered out. We aim to improve the precision of VRD and produce more relevant relations, therefore we focus on results with few relations. Table 3.4.2 displays results for 10, 20 and 100 predictions per image.

First, on VG-IMP [158], FOCUSEDVRD provides a significant improvement in the class macro recall, from 37 9% to 44.4% and a 3 points improvement over our baseline. However, this does not translate into the image macro recall, where we notice a drop from 88.7% to 87.7% for R@100 but a similar recall for fewer predictions. This suggests that the number of selected pairs is too low for a high number of predictions.

On VG-RMATTERS The last two columns show the number of pairs in the output Scene Graph at 10 and 20 predictions. Our model is able to keep a competitive recall with fewer selected pairs, which shows that it is able to more reliably choose which pairs to annotate than MOTIFNET.

Figure 5.3.2 shows the differences in recall per class of relation for MOTIFNET [166] and FOCUSEDVRD on VG-RMATTERS sorted by decreasing frequency in the training set. It shows that the increase in class macro recall is caused by the increase in recall of most classes, especially for uncommon relations.

	SgCls			RelCls		
	ImMacro			ImMacro		
	R@20	R@50	R@100	R@20	R@50	R@100
Freq Baseline	24.0	31.0	43.9	52.9	69.8	80.0
Pixel2Graph [102]	-	35.7	38.4	-	**82.0**	86.4
IMP [158]	-	43.4	47.2	-	75.2	83.6
SGP [58]	-	**45.5**	**50.8**	-	80.8	88.2
MotifNet [166]	37.6	44.5	47.7	66.6	81.2	88.3
Baseline (Ours)	37.1	44.1	47.2	**66.7**	81.4	**88.7**
FocusedVRD (Ours)	36.8	43.8	47.0	66.6	81.0	87.7

	RelCls		
	ClsMacro		
	R@20	R@50	R@100
MotifNet [166]	15.8	27.7	38.1
Baseline (Ours)	17.6	30.2	41.3
FocusedVRD (Ours)	**18.8**	**32.3**	**44.4**

Table 5.3.1 – Recall of FocusedVRD on VG-IMP. The relevance factor increases the ClsMacro recall at all sizes of scene graphs but slightly decreases the ImMacro recall.

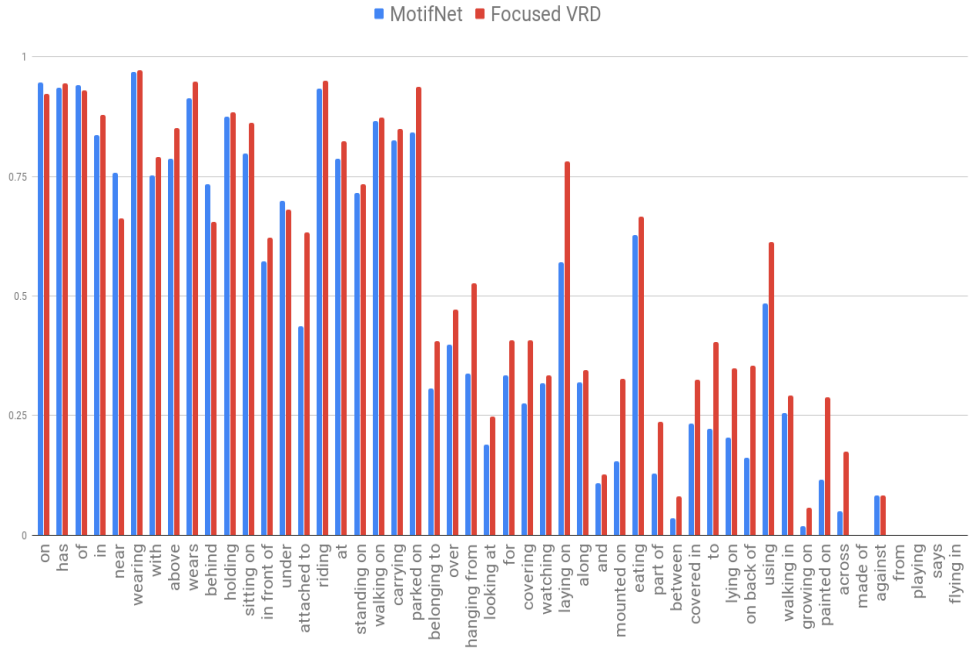


Figure 5.3.2 – Recall per relation class for MotifNet [166] and FocusedVRD (Ours) on VG-RMatters. By focusing predictions on most important object pairs, FocusedVRD is able to predict a more diverse set of relations thus increasing the recall of rarer relations while keeping a competitive global recall. Relations are ordered by decreasing frequency in the training set.

Ablation study Table 5.3.2 shows the results of our model on VG-RMatters and the influence of Relevance Modeling, with a Relevance Classifier (RC) and Binary Potential (BP). The network without RC or BP is the Baseline described in Chapter 3. Additionally, we compare them to our baseline with a relevance randomly sampled from a uniform distribution on $[0, 1]$. By adding a relevance factor to the Scene Graph extraction algorithm, class and image macro recalls of smaller scene graphs improve. This shows the importance of selecting relevant object pairs and that the improvement brought by the model is greater than by applying a random score to each object pair. The combination of both factors acts as ensembling by improving the selection of object pairs. The decrease in performance at 100 predictions, with RC or BP, shows that both methods are necessary to select most relevant pairs.

		SgCls			RelCls		
		ImMacro			ImMacro		
		R@10	R@20	R@100	R@10	R@20	R@100
Freq Baseline		18.5	28.4	50.9	25.9	40.7	77.4
Rand. Relevance		25.9	37.1	49.2	38.5	55.9	85.4
MotifNet [166]		26.3	38.0	55.5	39.6	58.4	87.8
RC	BP						
		26.7	38.5	56.3	40.3	58.9	**88.3**
✓		27.8	39.8	**56.4**	**44.2**	**62.3**	87.7
	✓	28.9	40.3	56.2	43.2	61.4	87.9
✓	✓	**29.4**	**41.0**	**56.4**	44.0	**62.3**	**88.3**

		RelCls			N pairs	
		ClsMacro				
		R@10	R@20	R@100	10	20
Freq Baseline		7.6	14.2	42.7	9.3	17.1
MotifNet [166]		11.8	19.8	46.6	8.6	14.6
RC	BP					
		12.6	21.1	47.8	8.6	14.6
✓		**14.5**	**24.0**	**52.6**	7.4	11.5
	✓	14.2	23.4	51.7	7.8	12.6
✓	✓	14.3	23.9	52.4	7.6	12.1

Table 5.3.2 – Ablation study on VG-RMatters. In the more balanced dataset, the relevance factor increases both ClsMacro and ImMacro recalls, especially for smaller scene graphs.

Figure 5.3.3 shows the output of FocusedVRD with RC and BP compared to the baseline and the ground truth. It shows the model's ability to detect and focus on important object pairs, removing the false relation (*flower*, *on*, *chair*). However, this also increases the risk of predicting false relation, such as (*chair, sitting on, table*). Removing these false positives would require a better classification of relations and/or probability thresholding. Other examples of ouptut with and without relevance are shown in Appendix B.

Figure 5.3.4 shows the histogram of estimated relevance, with Relevance Classification, Binary Potentials and the average, for object pairs with an annotated relation (top) and without (bottom). Since predictions are made by selecting top scoring relations, it is important to study the differences in scores between positive and negative relations. For RC, the ratio between the mean relevance probability for positive and negative relations is around 10, whereas it is around 4 for BP. This explains why fewer pairs are annotated when using RC.

In the case of RelCls, this allows the model to predict several relations for the most relevant pairs even for small scene graphs and thus increase recall, however it also increases the risk of missing less relevant pairs in the case of big scene graphs, where we see that RC has a lower recall than BP.

For SgCls, we observe the reverse phenomenon, which suggests that for smaller graphs, it is more important to have predictions less focused on a small number of pairs, in order to increase the probability to annotate pairs with the correct object classes.

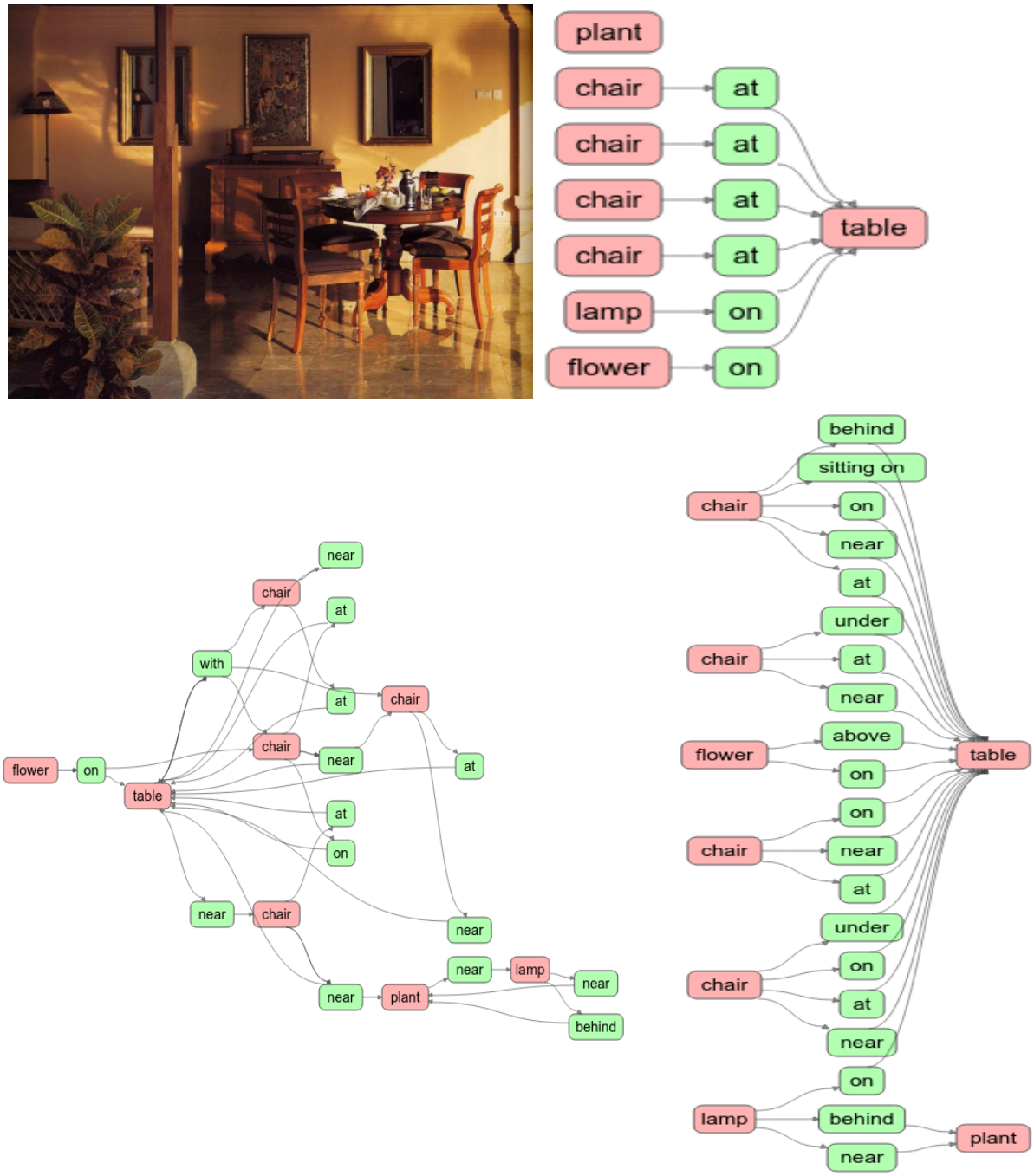
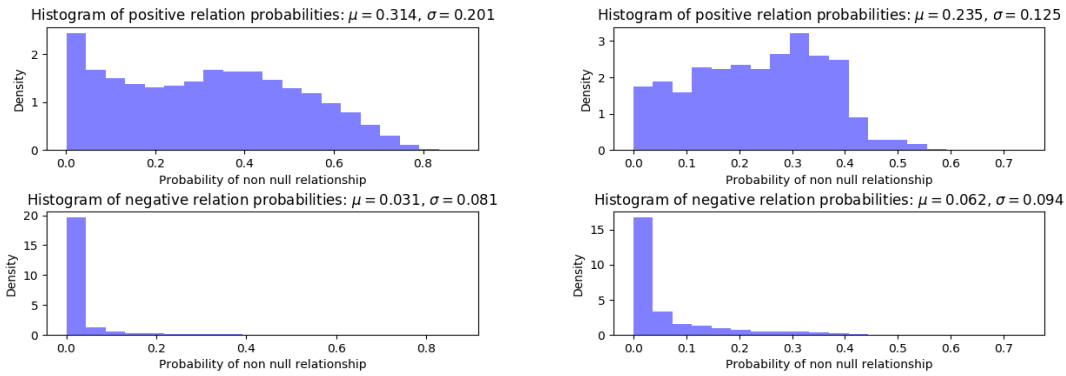
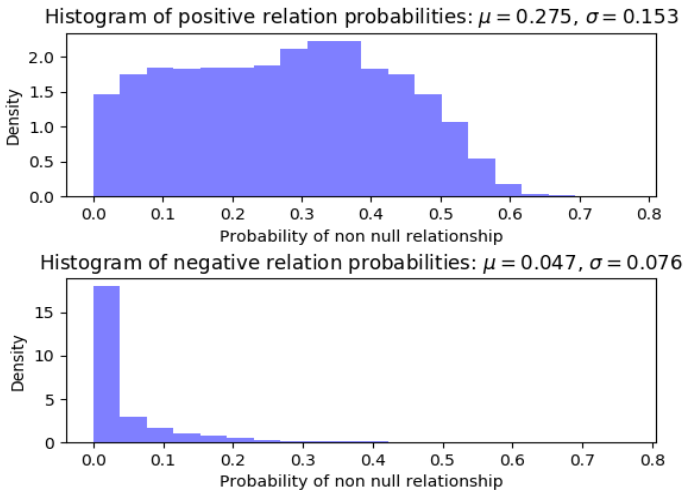


Figure 5.3.3 – Image example from VG and associated scene graph (*top right*), with extracted scene graphs of our baseline (bottom left) and FocusedVRD (RC+BP) (*bottom right*).



(a) RC (b) BP



(c) RC+BP

Figure 5.3.4 – Estimated Probabilities that any relation is annotated from Relevance Classifier, Prior Potential and the combination.

5.3.4 Discussion

The task of predicting the relevance of a relation is made difficult by the noise of the relevance ground truth, as it depends on many factors, such as the size, location, distances, salience of objects and the annotator.... Indeed, only 47% of the descripted regions of VG [71] are annotated with a relation, which suggests that the relations deemed relevant to a human being can be very different to another human being and makes the signal during training very noisy. Hence our choice to augment the learned classifier with binary potentials. Results in Tables 5.3.1 and 5.3.2 show that our method provides significant improvement to the class macro recall. These improvements translate into significant improvement of the image macro recall in the case of balance datasets such as VG-RMatters, especially for small scene graphs. Therefore the choice of incorporating this factor to the scene graph generation algorithm depends on the nature of the dataset under study. The differences in class macro Recall show that in both cases, when relevant classes are rare, *e.g.* in the case of limited training data, it improves relation detection. Finally, the small difference in performance for the frequency baseline between both splits at 100 predictions suggests that allowing bigger scene graphs provides a limited understanding of the model capabilities.

5.4 Conclusion

To sum up, we presented a novel relation detection method based on the estimation of the relevance of relations. A deep neural network is trained to predict the relevance of an object pair using annotated data. This relevance classifier is added to a statistics-based prior potential, measured on the training set. We integrate this new factor to the scene graph generation problems, which focuses predictions on fewer relevant pairs. We show that this allows the neural network to significantly increase the recall of uncommon relation classes, as the recall of several relations is increased by more than 100%, such as *across, painted on, on back of.* Our model with a modified scene graph generation process is able to better handle the diversity of relations than MotifNet, increasing the class macro R@20 by 21% and the image macro R@20 by 7%. This is especially true for smaller scene graphs, and helps decrease the output noise by making smaller scene graphs more exhaustive. This validates the proposed approach, showing that it is able to estimate the relevance of on object pair.

Chapter 6

Conclusion and perspectives

We conclude this work by summarizing the main contributions of this work, highlighting their advantages and limititations, then propose directions for further research in this field.

6.1 Main Contributions and Associated Perspectives

Visual Relation Detection is an important step in image understanding and depends highly on the quality of relation representations. This representation in turn depends on the quality of object representations as well as how their spatial, visual and semantic representations interact. As we showed in Chapter 1, relations representations are highly dependent on the visual representations of objects. In addition to this, the dependencies between relation classes makes a large number of training examples necessary for reliable classification and detection in the context of supervised learning.

Benchmark quality In Chapter 3, we showed that the combinatorial nature of VRD and a phenomenon which we call the Human Reporting Bias, make the number of available examples very low for a high number of classes and highly skewed towards a small number of classes. This prevents State of the Art models from being able to reliably detect uncommon relations. Furthermore, this also prevents existing benchmarks from estimating models how these models would perform on other benchmarks, because the trained models are over-exposed to a small number of possibilities. Hence we proposed a new metric on which to evaluate these models, focusing on the ability of the model to detect a large variety of relations. Finally, we proposed a new split of Visual Genome, which increases the variety of relations the most frequent pairs of objects.

Proposed approaches We have extended several methods used for semantic modelling to the case of Visual Relation Detection. We showed that modelling the semantic relationships between relations alleviates the need in training examples, improving the recall of rarer classes.

Finally, we proposed two novel approaches to modelling the relevance of a relation. We showed that integrating a new relevance factor into the Scene Graph generation process makes predictions more varied, increasing the recall of uncommon classes.

These contributions achieve competitive results on several datasets as shown on Table 6.1.1.

Dataset	Task	State of Art	Contribution	Results	Gain
VRD-set [89]	RelCls R@20	81.9 [165]*	SK	80.8	-1%
VG-IMP [158]	SGCls R@20	37.6 [166]	SK	37.2	-1%
VG-IMP [158]	RelCls R@20	66.6 [166]	Baseline	66.7	<0.1%
VG-Large	RelCls R@50	22.7 [158]**	SK + Relevance	45.2	99%
VG-RMatters	SGCls R@20	38.0 [166]**	Relevance	41.0	8%
VG-RMatters	RelCls R@20	58.4 [166]**	Relevance	62.3	7%
ClsMacro					
VG-IMP [158]	RelCls R@100	38.1 [166]	Relevance	44.4	17%
VG-RMatters	RelCls R@20	19.8 [166]	Relevance	23.9	21%

Table 6.1.1 – All metrics are computed on scene graphs without graph constraints. * are reimplementations of existing works. ** are computed using implementations made available by authors.

6.1.1 Human Reporting Bias in Relation Annotations

Contribution Humans choose to annotate some object pairs with some relations in a manner which tends to favor pairs of object categories as well as relations over others. This hinders the training of VRD models by increasing variance and making relation examples highly skewed. This motivated us to focus the study of the performance of our proposed models on less common relations. For this, we proposed a competitive VRD baseline and showed limitations of the standard training method of VRD models. We proposed a new metric that equally weighs all relation classes and a new split of Visual Genome, aimed at focusing on uncommon relations. We showed that an increase in this metric translated into better overall results on the latter split. Finally, we showed that performance on large scene graphs makes the task too easy as simple baselines are able to get competitive results, thus we focus on results with smaller ones.

The proposed baseline outperformed the current State of the Art method on RelCls, MotifNet [166], by 0.5% on VG-IMP [71, 158] and 1.3% on VG-RMatters (Ours).

Perspectives The interaction between the semantic, visual and spatial modalities has been studied by concatenating the three representations or, as in our Baseline, similarly to [41, 170], using early fusion. These modalities have varied importance for the studied relation classes, and further exploration of how they interact and how each feature is relevant to given relation instances could help learning to separate those classes. An attention mechanism could be applied to the extracted features, in the same vein as in [58, 154, 158].

Decreasing the cross-entropy loss does not correlate with improved performance

on the test metrics, which require that relations be ranked by the combined scores of several classification branches. Several works [82, 84] have tackled this problem by proposing new methods, based on Reinforcement Learning or training with a ranking-based loss. One avenue of research would consist of providing more complete annotations, for potentially fewer images, proposing several relations for each object pair and changing the evaluation metric to mean Average Precision.

Finally, given a more balanced test set, such as VG-RMatters, several methods aimed at learning in imbalanced settings could be used in order to improve the training of uncommon relations, such as negative example sampling or focal loss [87].

6.1.2 Overcoming Relation Imbalance with Semantic Modelling

Contributions The traditional classification framework is hindered by the previously mentioned issues. Learning representations of relations in a metric space has proven beneficial. Indeed, they result in more interpretable models, which showed further limitations of the joint modelling of the three modalities and the trade-off between common-sense knowledge and visual data. Furthermore, it enabled predictions based on previously seen relations and semantic consistency. Finally it proved very promising in predicting a larger variety of relations. This improved the recall by 1.5% on VRD-set when compared to our baseline and the class macro recall by 12.7%.

The comparison with two methods of explicit modelling of synonyms highlights that implicit modelling of synonyms has the advantage of being adapted to the visual domain, where visual relations have many different meanings and are not synonym to the same relations as text relations. The rule distillation method presented by [62], for which we have devised rules based on the similarity between pre-trained word embeddings (**SK**), has been proven beneficial on this task, allowing the model to use external text data in a robust way, ignoring data that does not coincide with visual data. This had a significant impact on the dataset with a large number of relations, VG-Large, improving the recall by 33%.

Perspectives. Our work on Semantic Modelling, motivated by the need to forgo the assumption that relations are exclusive, is related in this way to other works which completely change the training process [84, 89, 165, 169]. Modelling hierarchical relations between object and relation classes is a natural step and has been shown beneficial in [164], but it was limited to what could be translated to data augmentation. This could be further studied by integrating it into Graph Convolutional Networks [68, 125]. Furthermore, other advances made in metric learning, such as

hyperbolic embeddings [104], have shown to allow effective prediction of missing links in knowledge graphs, such as WordNet [34] and could either allow completing semantic networks or allow representations of relations to be more consistent with these networks.

6.1.3 Relation Relevance

Contributions The observation that humans naturally filter out sensory information in order to focus on the most relevant pieces of information motivated us to propose a new scene graph generation process. This process introduces the new relevance factor which weighs relations and helps filter out irrelevant ones. Estimating the relevance of a relation is difficult because it depends on a large variety of factors, from object dependent to annotator dependent factors. Our first new approach, which aimed to estimate this using self-supervision with image topic did not provide significant improvements on Visual Genome. We then proposed to estimate this relevance with the trained "*background*" classifier of the model and average it with a statistics-based potential. This potential leverages the available data from the training set and smooths the relevance estimation. This has proven very effective at increasing the recall of small scene graphs and the variety of predicted relations, increasing the Image Macro recall by 7% and the Class Macro Recall by 13%. On VG-Large, we note that the combination of Relevance and Semantic Distillation bring a 99% improvement in recall when compared to IMP [158], the state of the art method at the times. This shows the relevance of both approaches, but we note that this dataset has not been widely studied, which also explains the large difference. The highest improvements in this work are brought about by this approach. This highlights the importance on focusing on the Scene Graph generation process, a facet of Visual Relation Detection which has been scarcely studied.

Perspectives Several methods could be used to reduce the impact of the noise in the relevance variable. The first, following [99], would consist in jointly modelling the relevance variable and the concept variable. Similarly to relation classification, relation detection is hard due to the high imbalance between negative and positive examples. This could be targetted by methods aimed at tackling this problem such as example sampling (Section 2.5) or knowledge distillation (Section 2.4.2).

Furthermore, given a dataset with several annotated relations per object pair, we could train a regression model that estimates the number of relations to predict for each object pair, given their spatial, semantic and visual representations. While TopicNet has not proven effective at modelling the relevance of relations, Woo *et al.* [154] showed an attention mechanism between objects is able to capture the

presence of a relevant object pair. Further comparisons between this work and our relevance estimation could provide new avenues of research.

6.2 Future Directions

Chapter 5 has proven the impact of the scene graph generation process on the performance of the model. The differences between the training and inference evaluations have received little attention, except in [84] which learns which object pairs to annotated by Reinforcement Learning, and to a lesser extent in [82]. The estimation of the relation relevance has received an important focus in this work but remains an open question, as the results show limitations. Furthermore, instead of a ranking-based process, sampling relations from the estimated distribution could be introduced into the training process, tackling the inadequacy of the training loss for the current test setting.

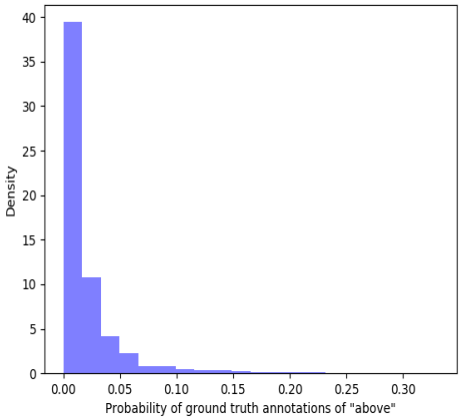
Furthermore, while our proposed methods have provided improvements on the relation classification on several datasets, we still lack understanding of how the features of the spatial and visual modalities interact and how they impact the model predictions. As discussed in the Introduction of this work, this issue is an important one, because models tend to latch onto spurious correlations [8, 31, 45, 121] . This might be reinforced by the intrinsic biases of the data, where a limited number of relations can be true for a given object pair. Our work on metric learning for VRD allowed us to get a better interpretation of the model but our understanding is still lacking.

Finally, Visual Genome has opened the research in supervised learning, by providing a large number of annotated images. However, this dataset is imbalanced and limited to a small number of relations with sufficient examples. The annotation of such a dataset would be very time consuming, thus existing datasets such as Visual Genome and VRD-set could be complemented with web content, which is more noisy. The commonly used metrics also provide a limited understanding of model capabilities. We suggest that the test setting in itself might not be the most adapted setting to evaluate VRD predicate models. Indeed, the combinatorial nature of the problem makes it very hard to obtain complete annotations. Thus, as in [64], we suggest to evaluate them on more downstream tasks, such as image retrieval, image generation, caption generation, Visual Question Answering, which could also contribute in showing the impact of this task on other applications.

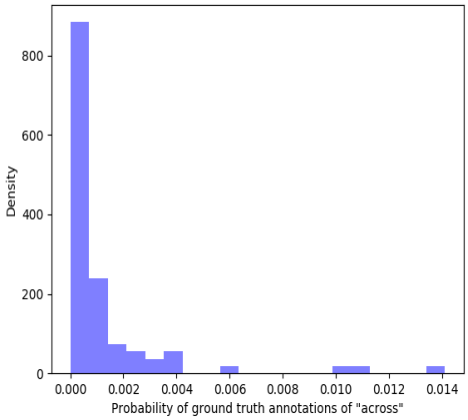
Appendices

Appendix A

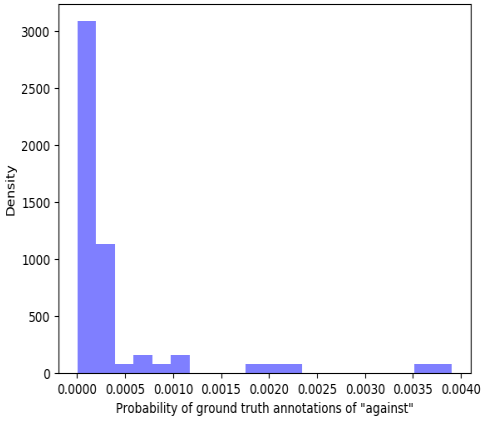
Human Reporting Bias and Class Output Probabilities



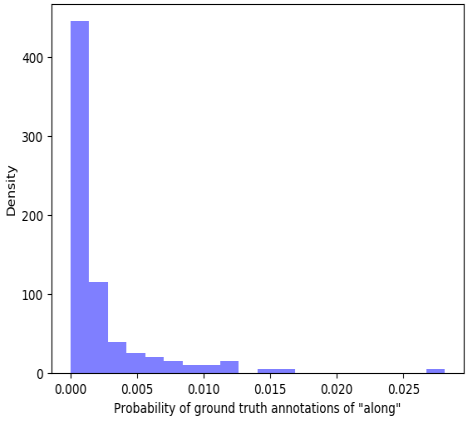
above



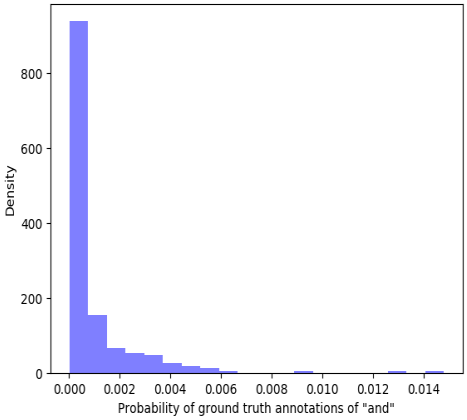
across



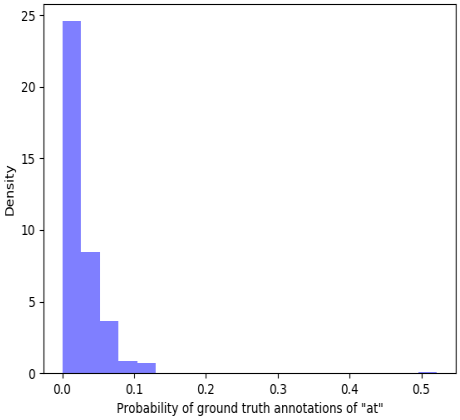
against



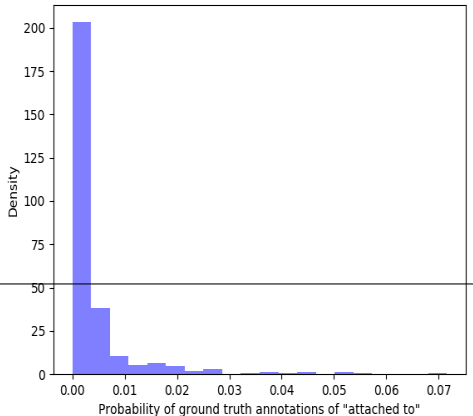
along

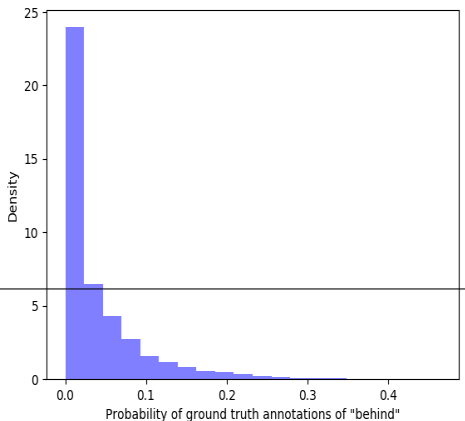


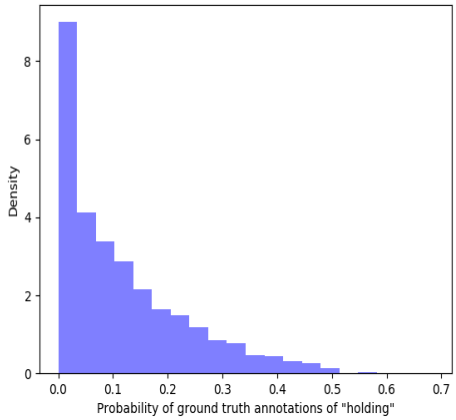
and



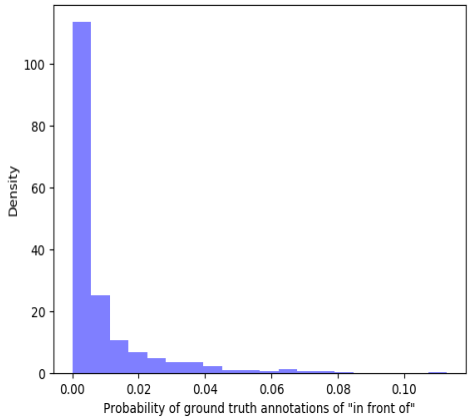
at



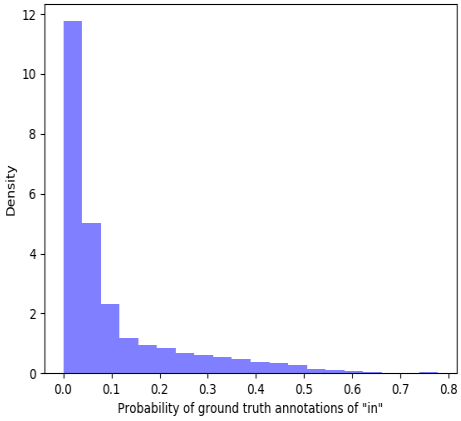




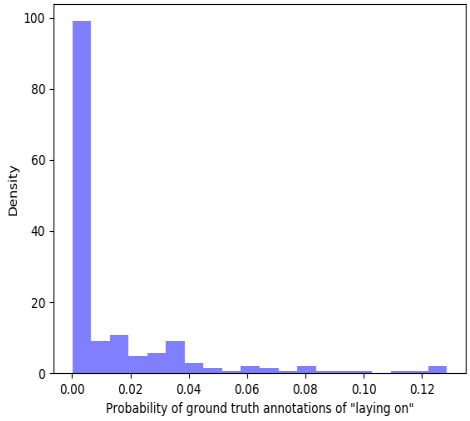
holding



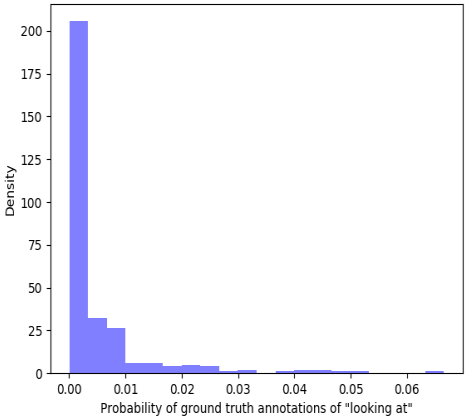
in front of



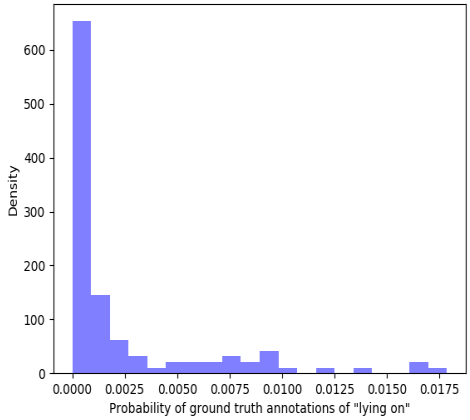
in



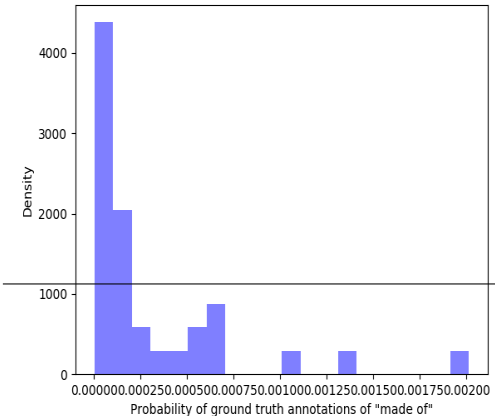
laying on

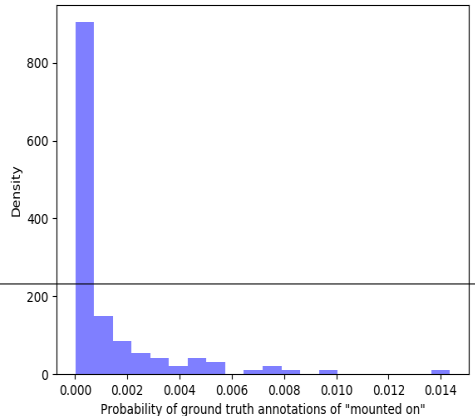


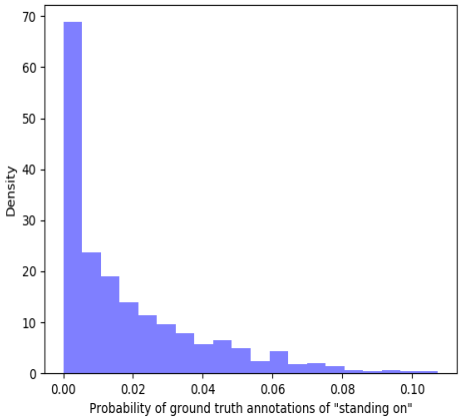
looking at



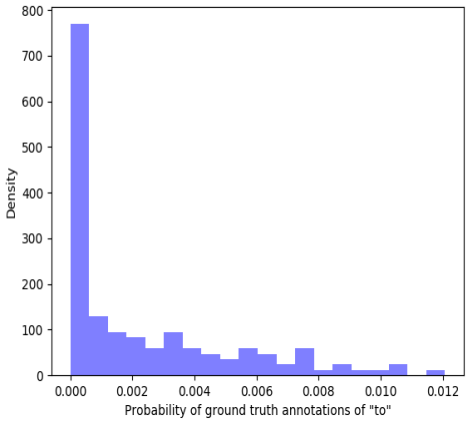
lying on



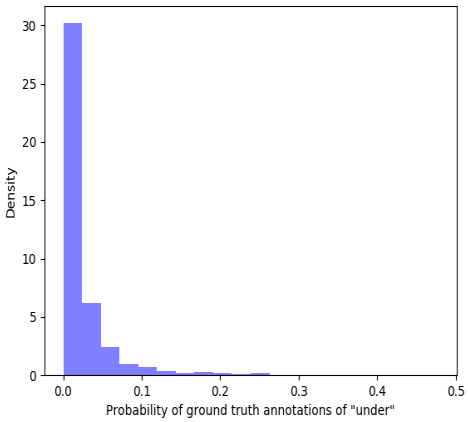




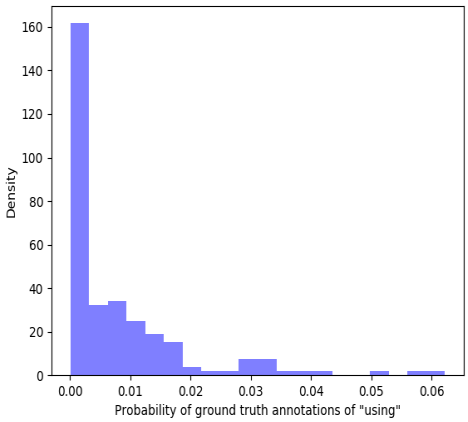
standing on



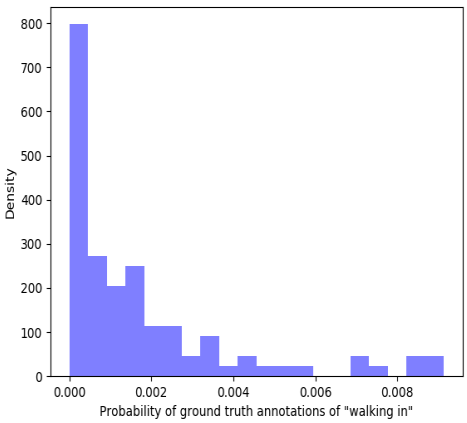
to



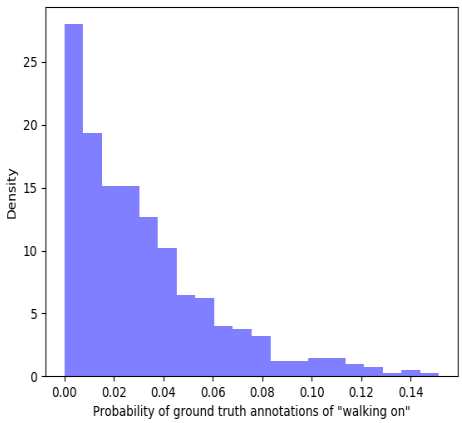
under



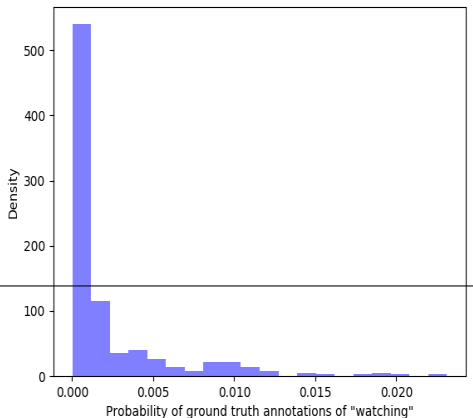
using

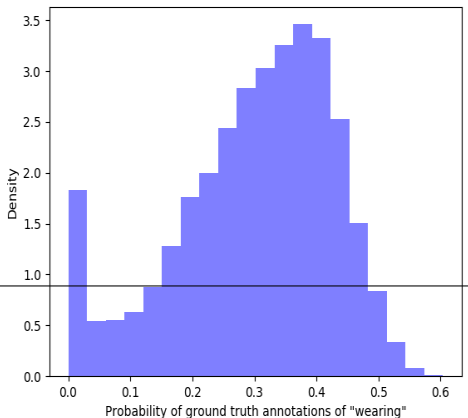


walking in



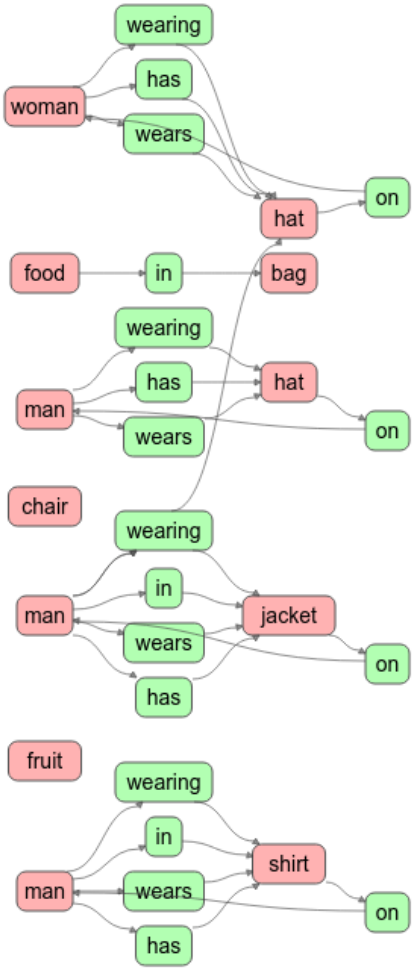
walking on



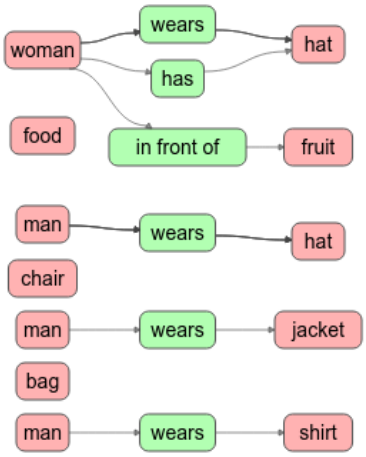


Appendix B

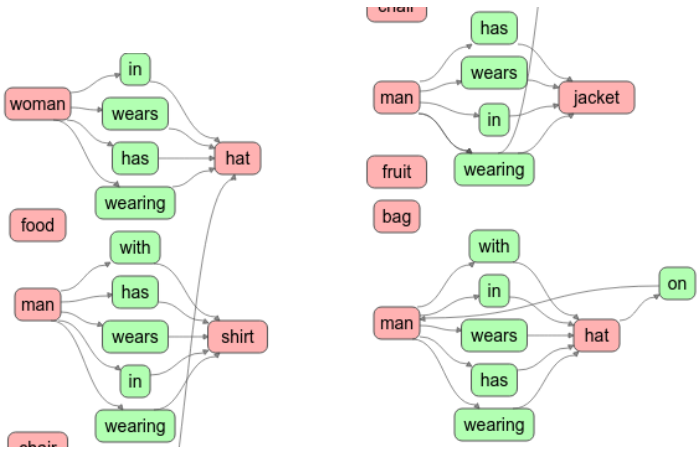
Output examples with relevance



(a) Baseline Scene graph

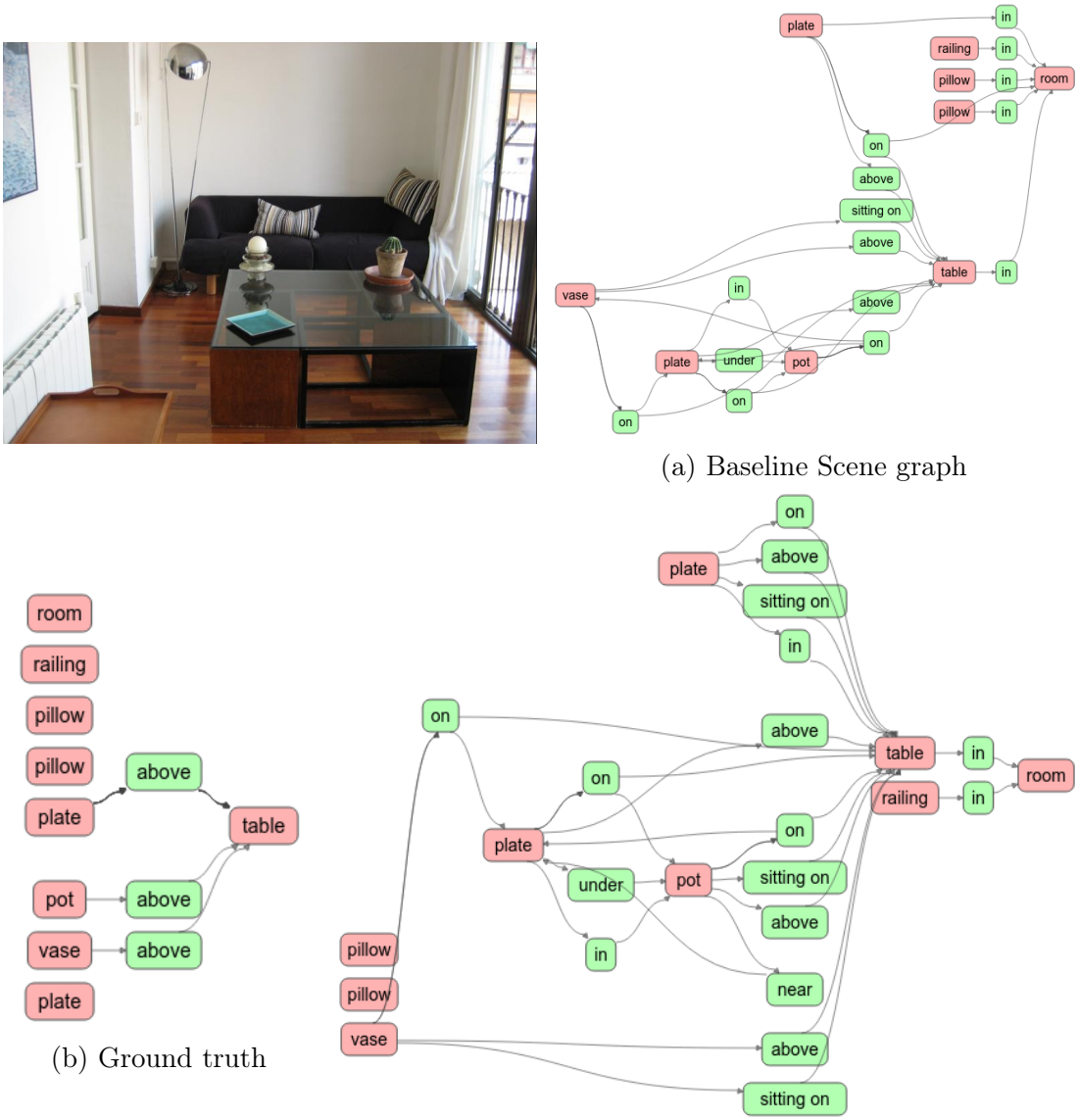


(b) Ground truth



(c) Scene graph with relevance

Figure B.0.1 – The relevance factor removes several (*clothing, person*) relations, increasing the number of synonyms of *wear* for (*person, clothing*)

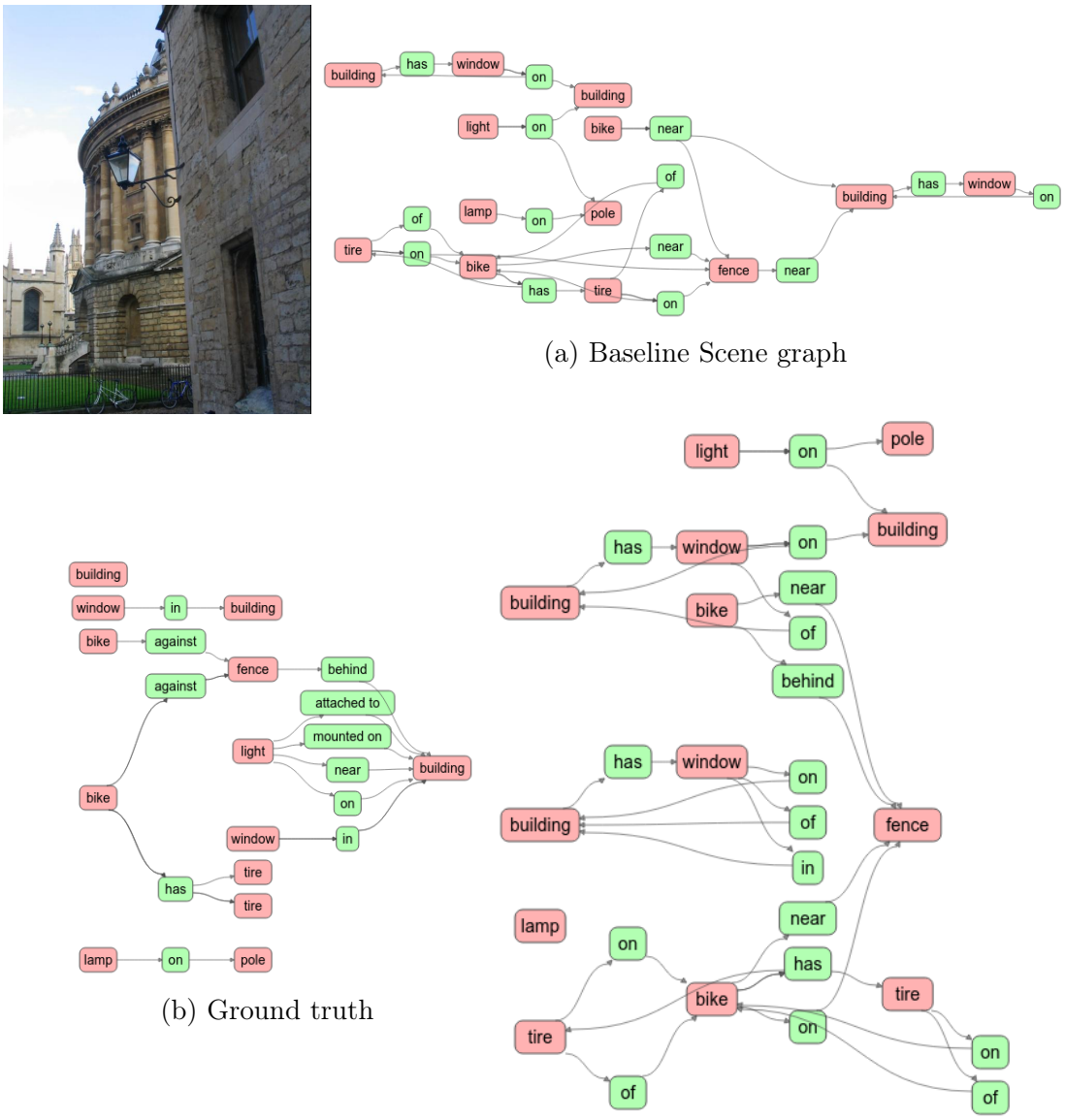


(a) Baseline Scene graph

(b) Ground truth

(c) Scene graph with relevance

Figure B.0.2 – The relevance factor removes some (*artifact, room*) relations and adds the relation (*pot, above, table*).



(a) Baseline Scene graph

(b) Ground truth

(c) Scene graph with relevance

Figure B.0.3 – The relevance factor removes a true positive: (*lamp, on pole*).

Bibliography

[1] Processing Power Compared. URL https://pages.experts-exchange.com/processing-power-compared.

[2] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering. In *CVPR*, 2018.

[3] Zinonas Antoniou. *Real-Time Adaptation to Time-Varying Constraints for Reliable mHealth Video Communications*. PhD thesis, 2017.

[4] Rick Armbrust. Capturing Growth: Photo Apps and Open Graph. Technical report, 2012. URL https://developers.facebook.com/blog/post/2012/07/17/capturing-growth--photo-apps-and-open-graph/.

[5] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. In *ICLR*, 2015.

[6] Nicolas Ballas. *Modélisation de contextes pour l'annotation sémantique de vidéos*. PhD thesis, 2014. URL https://pastel.archives-ouvertes.fr/pastel-00958135.

[7] Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal Machine Learning: A Survey and Taxonomy. *Pattern Analysis and Machine Intelligence*, 2017.

[8] Yonatan Belinkov, Adam Poliak, Stuart M Shieber, Benjamin Van Durme, and Alexander M Rush. Don't Take the Premise for Granted: Mitigating Artifacts in Natural Language Inference. In *ACL*, 2019.

[9] Alexander C Berg, Tamara L Berg, Hal Daume, Jesse Dodge, Amit Goyal, Xufeng Han, Alyssa Mensch, Margaret Mitchell, Aneesh Sood, Karl Stratos, and Kota Yamaguchi. Understanding and Predicting Importance in Images. In *CVPR*, 2012.

[10] David M Blei, Andrew Y Ng, and Jordan@cs Berkeley Edu. Latent Dirichlet Allocation Michael I. Jordan. *J. Mach. Learn. Res.*, 3, 2003.

[11] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. End to End Learning for Self-Driving Cars. 2016.

[12] Thorsten Brants and Alex Franz. Web 1t 5-gram version 1. *Linguistic Data Consortium*, 2006.

[13] Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 2018.

[14] Yu Wei Chao, Zhan Wang, Yugeng He, Jiaxuan Wang, and Jia Deng. HICO: A benchmark for recognizing human-object interactions in images. In *ICCV*, 2015. doi: 10.1109/ICCV.2015.122.

[15] Nitesh Chawla, Kevin Bowyer, Lawrence Hall, and Philip Kegelmeyer. SMOTE: Synthetic Minority Over-sampling Techniqu. *Journal of Artificial Intelligence Research*, 2002. URL `https://jair.org/index.php/jair/article/view/10302/24590`.

[16] Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. Learning Efficient Object Detection Models with Knowledge Distillation. In *NIPS*, 2017.

[17] Liang Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *PAMI*, 2018. doi: 10.1109/TPAMI.2017.2699184.

[18] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *EMNLP*, 2014.

[19] Noam Chomsky. Three models for the description of language. *IRE Transactions on Information Theory*, 1956. ISSN 21682712. doi: 10.1109/TIT.1956.1056813.

[20] Aidan Clark, Jeff Donahue, and Karen Simonyan. Efficient Video Generation on Complex Datasets. 2019.

[21] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *RecSys 2016 - Proceedings of the 10th ACM*

Conference on Recommender Systems, 2016. ISBN 9781450340359. doi: 10.1145/2959100.2959190.

[22] Yin Cui, Feng Zhou, Yuanqing Lin, and Serge Belongie. Fine-grained Categorization and Dataset Bootstrapping using Deep Metric Learning with Humans in the Loop. In *CVPR*, 2016. ISBN 1512.05227v2.

[23] Bo Dai, Yuqi Zhang, and Dahua Lin. Detecting Visual Relationships with Deep Relational Networks. In *CVPR*, 2017. doi: 10.1109/CVPR.2017.352.

[24] Jifeng Dai, Kaiming He, and Jian Sun. BoxSup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *ICCV*, 2015. ISBN 9781467383912. doi: 10.1109/ICCV.2015.191.

[25] Jia Deng, Jonathan Krause, Alexander C Berg, and Li Fei-Fei. Hedging Your Bets: Optimizing Accuracy-Specificity Trade-offs in Large Scale Visual Recognition. In *CVPR*, 2012.

[26] Jia Deng, Nan Ding, Yangqing Jia, Andrea Frome, Kevin Murphy, Samy Bengio, Yuan Li, Hartmut Neven, and Hartwig Adam. Large-Scale Object Classification Using Label Relation Graphs. In *European Conference on Computer Vision*, 2014.

[27] Jia Deng Jia Deng, Wei Dong Wei Dong, R. Socher, Li-Jia Li Li-Jia Li, Kai Li Kai Li, and Li Fei-Fei Li Fei-Fei. ImageNet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2–9, 2009. ISSN 1063-6919. doi: 10.1109/CVPR.2009.5206848.

[28] Thomas G Dietterich. Ensemble Methods in Machine Learning. In *International workshop on multiple classifier systems*, 2000. URL `http://www.cs.orst.edu/~tgd`.

[29] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised Visual Representation Learning by Context Prediction. In *ICCV*, 2015.

[30] Alexey Dosovitskiy, Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative Unsupervised Feature Learning with Convolutional Neural Networks. In *NIPS*, 2014.

[31] Logan Engstrom, Andrew Ilyas, Aleksander Madry, Shibani Santurkar, Brandon Tran, and Dimitris Tsipras. A Discussion of 'Adversarial Examples Are

Not Bugs, They Are Features': Discussion and Author Responses. *Distill*, 219. doi: 10.23915/distill.00019.7. URL `https://distill.pub/2019/advex-bugs-discussion/`.

[32] Mark Everingham, Luc Van Gool, Christopher K I Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010. ISSN 09205691. doi: 10.1007/s11263-009-0275-4.

[33] Yuan Fang, Kingsley Kuan, Jie Lin, Cheston Tan, and Vijay Chandrasekhar. Object Detection Meets Knowledge Graphs. *IJCAI*, pages 1661–1667, 2017.

[34] Christiane Fellbaum. *WordNet: An Electronic Lexical Database*, volume 71. Bradford Books, 1998. ISBN 026206197X. doi: 10.1139/h11-025.

[35] Pedro F Felzenszwalb, Ross B Girshick, David Mcallester, and Deva Ramanan. Object Detection with Discriminatively Trained Part Based Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.

[36] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. Self-Supervised Video Representation Learning With Odd-One-Out Networks. In *ICCV*, 2017.

[37] R. A. FISHER. THE USE OF MULTIPLE MEASUREMENTS IN TAXONOMIC PROBLEMS. *Annals of Eugenics*, 1936. doi: 10.1111/j.1469-1809.1936.tb02137.x.

[38] Andrea Frome, Greg S Corrado, Jonathon Shlens, Samy Bengio Jeffrey Dean, Aurelio Ranzato, and Tomas Mikolov. DeViSE: A Deep Visual-Semantic Embedding Model. In *NIPS*, 2013.

[39] Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. Multimodal Compact Bilinear Pooling for Visual Question Answering and Visual Grounding. *Arxiv*, 6 2016. URL `http://arxiv.org/abs/1606.01847`.

[40] Carolina Galleguillos, Andrew Rabinovich, and Serge J. Belongie. Object categorization using co-occurrence, location and appearance. In *CVPR*, 2008. ISBN 9781424422432. URL `https://doi.org/10.1109/CVPR.2008.4587799`.

[41] Chen Gao, Yuliang Zou, and Jia-Bin Huang. iCAN: Instance-Centric Attention Network for Human-Object Interaction Detection. In *BMVC*, 2018. URL `https://gaochen315.github.io/iCAN/`.

[42] Mor Geva, Yoav Goldberg, and Jonathan Berant. Are We Modeling the Task or the Annotator? An Investigation of Annotator Bias in Natural Language Understanding Datasets. In *EMNLP-IJCNLP*, 2019.

[43] Golnaz Ghiasi and Charless C. Fowlkes. Laplacian pyramid reconstruction and refinement for semantic segmentation. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016. ISBN 9783319464862. doi: 10.1007/978-3-319-46487-9{_}32.

[44] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. UNSUPERVISED REPRESENTATION LEARNING BY PRE-DICTING IMAGE ROTATIONS. In *ICLR*, 2018.

[45] Justin Gilmer and Dan Hendrycks. A Discussion of 'Adversarial Examples Are Not Bugs, They Are Features': Adversarial Example Researchers Need to Expand What is Meant by 'Robustness'. *Distill*, 2019. doi: 10.23915/distill.00019.1. URL `https://distill.pub/2019/advex-bugs-discussion/response-1`.

[46] Rohit Girdhar and Deva Ramanan. Attentional Pooling for Action Recognition. In *NIPS*, 2017.

[47] Ross Girshick. Fast R-CNN. In *ICCV*, 2015.

[48] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 11 2014. URL `http://arxiv.org/abs/1311.2524`.

[49] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014. ISSN 10636919. doi: 10.1109/CVPR.2014.81.

[50] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010.

[51] Jacob Goldberger, Sam Roweis, Geoff Hinton, and Ruslan Salakhutdinov. Neighbourhood Components Analysis. In *NIPS*, 2004. URL `https://www.cs.toronto.edu/~hinton/absps/nca.pdf`.

[52] Lluis Gomez, Yash Patel, Marcal Rusinol, Dimosthenis Karatzas, and C V Jawahar. Self-supervised learning of visual features through embedding images into text topic spaces. In *CVPR*, 2017.

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

[54] Kyle Gorman and Steven Bedrick. We need to talk about standard splits. In *ACL*, 2019.

[55] Yash Goyal, Tejas Khot, Aishwarya Agrawal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. *International Journal of Computer Vision*, 2019. ISSN 15731405. doi: 10.1007/s11263-018-1116-0.

[56] Saurabh Gupta, Judy Hoffman, and Jitendra Malik. Cross Modal Distillation for Supervision Transfer. In *CVPR*, 2016. URL `https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Gupta_Cross_Modal_Distillation_CVPR_2016_paper.pdf`.

[57] Haibo He, Yang Bai, Edwardo A Garcia, and Shutao Li. ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning. In *IEEE International Joint Conference on Neural Networks*, 2008. ISBN 9781424418213.

[58] Roei Herzig, Moshiko Raboh, Gal Chechik, Jonathan Berant, and Amir Globerson. Mapping Images to Scene Graphs with Permutation-Invariant Structured Prediction. In *NIPS*, 2018.

[59] Geoffrey Hinton and Jeff Dean. Distilling the Knowledge in a Neural Network. In *NIPS Deep Learning Workshop*, 2014. URL `https://arxiv.org/pdf/1503.02531.pdf`.

[60] Sepp Hochreiter and Jürgen Schmidhuber. full-text. *Neural Computation*, 1997.

[61] Seunghoon Hong, Junhyuk Oh, Honglak Lee, and Bohyung Han. Learning Transferrable Knowledge for Semantic Segmentation with Deep Convolutional Neural Network. In *CVPR*, 2016.

[62] Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric Xing. Harnessing Deep Neural Networks with Logic Rules. In *ACL*, 2016. ISBN 9781510827585. doi: 10.18653/v1/P16-1228.

[63] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. .

[64] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David A Shamma, Michael S Bernstein, and Li Fei-Fei. Image Retrieval using Scene Graphs. Technical report, . URL `https://hci.stanford.edu/publications/2015/scenegraphs/JohnsonCVPR2015.pdf`.

[65] Kaiser, Ofir Nachum, Aurko Roy, and Samy Bengio. Learning to Remember Rare Events. In *ICLR*, 2017.

[66] Will Kay, João Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The Kinetics Human Action Video Dataset. Technical report.

[67] J. Kiefer and J. Wolfowitz. Stochastic Estimation of the Maximum of a Regression Function. *The Annals of Mathematical Statistics*, 1952.

[68] Thomas N Kipf and Max Welling. Semi-supervised Classification with Graph Convolutioal Networks. In *ICLR*, 2017.

[69] Ryan Kiros, Ruslan Salakhutdinov, and Richard Zemel. Multimodal Neural Language Models. *Icml*, page 595–603, 2014. URL `http://www.jmlr.org/proceedings/papers/v32/kiros14.pdf`.

[70] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese Neural Networks for One-shot Image Recognition. Technical report. URL `https://www.cs.cmu.edu/~rsalakhu/papers/oneshot1.pdf`.

[71] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. 2016. doi: 10.1007/s11263-016-0981-7.

[72] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *Advances In Neural Information Processing Systems*, pages 1–9, 2012. ISSN 10495258. doi: http://dx.doi.org/10.1016/j.protcy.2014.09.007.

[73] H Kuehne, H Jhuang, E Garrote, T Poggio, and T Serre. HMDB: A Large Video Database for Human Motion Recognition. In *High Performance Computing in Science and Engineering*, 2012.

[74] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig,

and Vittorio Ferrari. IJCV submission in review The Open Images Dataset V4 Unified image classification, object detection, and visual relationship detection at scale. 2018.

[75] Brenden M Lake, Ruslan Salakhutdinov, Jason Gross, and Joshua B Tenenbaum. One shot learning of simple visual concepts. In *In {Proceedings of the 33rd Annual Conference of the Cognitive Science Society}*, 2011.

[76] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 2015.

[77] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building Machines That Learn and Think Like People. *Behavioral and Brain Sciences*, 2017.

[78] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2009*, 2009. ISBN 9781424439935. doi: 10.1109/CVPRW.2009.5206594.

[79] Claudia Leacock and Martin Chodorow. Filling in a sparse training space for word sense identification. 1994.

[80] Yikang Li, Wanli Ouyang, Xiaogang Wang, and Xiao'ou Tang. ViP-CNN: Visual Phrase Guided Convolutional Neural Network. In *CVPR*, 2017. doi: 10.1109/CVPR.2017.766.

[81] Yuncheng Li, Jianchao Yang, Yale Song, Yahoo Research, Liangliang Cao, Jiebo Luo, and Li-Jia Li. Learning from Noisy Labels with Distillation. In *ICCV*, 2017.

[82] Kongming Liang, Yuhong Guo, Hong Chang, and Xilin Chen. Visual Relationship Detection with Deep Structural Ranking.

[83] Kongming Liang, Yuhong Guo, Hong Chang, and Xilin Chen. Visual Relationship Detection with Deep Structural Ranking. In *AAAI*, 2018.

[84] Xiaodan Liang, Lisa Lee, and Eric P Xing. Deep Variation-structured Reinforcement Learning for Visual Relationship and Attribute Detection. In *CVPR*, 2017. doi: 10.1109/CVPR.2017.469. URL `https://arxiv.org/pdf/1703.03054.pdfhttp://arxiv.org/abs/1703.03054`.

[85] Guosheng Lin, Chunhua Shen, Anton Van Den Hengel, and Ian Reid. Efficient Piecewise Training of Deep Structured Models for Semantic Segmentation. In *CVPR*, 2016. ISBN 9781467388504. doi: 10.1109/CVPR.2016.348.

[86] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C Lawrence Zitnick, and Piotr Dolí. Microsoft COCO: Common Objects in Context. URL `https://arxiv.org/pdf/1405.0312.pdf`.

[87] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal Loss for Dense Object Detection. In *ICCV*, 2017.

[88] Ziwei Liu, Xiaoxiao Li, Ping Luo, Chen Change Loy, and Xiaoou Tang. Semantic image segmentation via deep parsing network. In *ICCV*, 2015. ISBN 9781467383912. doi: 10.1109/ICCV.2015.162.

[89] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In *ECCV*, 2016. ISBN 9783319464473. doi: 10.1007/978-3-319-46448-0{_}51.

[90] Haley MacLeod, Cynthia L. Bennett, Meredith Ringel Morris, and Edward Cutrell. Understanding Blind People's Experiences with Computer-Generated Captions of Social Media Images. 2017. doi: 10.1145/3025453.3025814.

[91] J Macqueen. Some methods for classification and analysis of multivariate observations. In *Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1967.

[92] Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B. Tenenbaum, and Jiajun Wu. The Neuro-Symbolic Concept Learner: Interpreting Scenes, Words, and Sentences From Natural Supervision. In *ICLR*, 2019.

[93] Yuzhao Mao, Chang Zhou, Xiaojie Wang, and Ruifan Li. Show and Tell More: Topic-Oriented Multi-Sentence Image Captioning. In *IJCAI*, 2018.

[94] Gary Marcus, I thank Christina, François Chollet, Ernie Davis, Zack Lipton, Stefano Pacifico, Suchi Saria, and Athena Vouloumanos. Deep Learning: A Critical Appraisal. Technical report. URL `http://www.nytimes.com/2012/11/24/science/scientists-see-advances-in-deep-learning-a-part-of-artificial-`.

[95] Kenneth Marino, Ruslan Salakhutdinov, and Abhinav Gupta. The More You Know: Using Knowledge Graphs for Image Classification. In *CVPR*, 2017. doi: 10.1109/CVPR.2017.10.

[96] R Thomas Mccoy, Ellie Pavlick, and Tal Linzen. Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference. In *ACL*, 2019.

[97] Tomas Mikolov, Greg Corrado, Kai Chen, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. In *ICLR*, 2013. ISBN 1532-4435. doi: 10.1162/153244303322533223.

[98] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and Learn: Unsupervised Learning using Temporal Order Verification. In *ECCV*, 2016. URL `https://arxiv.org/pdf/1603.08561.pdf`.

[99] Ishan Misra, C Lawrence Zitnick, Margaret Mitchell, and Ross Girshick. Seeing through the Human Reporting Bias: Visual Classifiers from Noisy Human-Centric Labels. In *CVPR*, 2016.

[100] Mathew Monfort, Alex Andonian, Bolei Zhou, Kandan Ramakrishnan, Sarah Adel Bargal, Tom Yan, Lisa Brown, Quanfu Fan, Dan Gutfruend, and Carl Vondrick. Moments in Time Dataset: one million videos for event understanding. *PAMI*, 2019.

[101] Arvind Neelakantan, Quoc V Le Google Brain, Martín Abadi Google Brain, Andrew McCallum, and Dario Amodei. LEARNING A NATURAL LANGUAGE INTERFACE WITH NEURAL PROGRAMMER. In *ICLR*, 2017.

[102] Alejandro Newell and Jia Deng. Pixels to Graphs by Associative Embedding. In *NIPS*, 2017.

[103] Alejandro Newell, Zhiao Huang, and Jia Deng. Associative Embedding: End-to-End Learning for Joint Detection and Grouping. In *NIPS*, 2017.

[104] Maximilian Nickel and Douwe Kiela. Poincaré Embeddings for Learning Hierarchical Representations. Technical report. URL `http://papers.nips.cc/paper/7213-poincare-embeddings-for-learning-hierarchical-representations.pdf`.

[105] Maximilian Nickel and Douwe Kiela. Poincaré Embeddings for Learning Hierarchical Representations. 2017. URL `https://arxiv.org/pdf/1705.08039.pdf`.

[106] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks. In *CVPR*, 2014.

[107] Vicente Ordonez, Jia Deng, Yejin Choi, Alexander C Berg, and Tamara L Berg. From Large Scale Image Categorization to Entry-Level Categories. In *ICCV*, 2013.

[108] Mark Palatucci, Geoffrey E. Hinton, Dean Pomerleau, and Tom M. Mitchell. Zero-Shot Learning with Semantic Output Codes. In *Advances in Neural Information Processing Systems 22 (NIPS 2009)*, 2009. ISBN 9781615679119.

[109] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context Encoders: Feature Learning by Inpainting. In *CVPR*, 2016.

[110] Karl Pearson. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 1901. doi: 10.1080/14786440109462720.

[111] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global Vectors for Word Representation. In *EMNLP*, 2014. ISBN 9781937284961. doi: 10.3115/v1/D14-1162. URL `http://www.aclweb.org/anthology/D14-1162http://aclweb.org/anthology/D14-1162`.

[112] Julia Peyre, Ivan Laptev, Cordelia Schmid, and Josef Sivic. Weakly-supervised learning of visual relations. In *ICCV*, 2017.

[113] Francois Plesse, Alexandru Ginsca, Bertrand Delezoide, and Francoise Prêteux. Visual Relationship Detection Based on Guided Proposals and Semantic Knowledge Distillation. In *ICME*, 2018.

[114] François Plesse, Alexandru Ginsca, Bertrand Delezoide, and Françoise Prêteux. Learning Prototypes for Visual Relationship Detection. In *CBMI*, 2018.

[115] Ning Qian. On the Momentum Term in Gradient Descent Learning Algorithms. Technical report. URL `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.57.5612&rep=rep1&type=pdf`.

[116] Ilija Radosavovic, Piotr Dollár, Ross Girshick, Georgia Gkioxari, and Kaiming He. Data Distillation: Towards Omni-Supervised Learning. In *CVPR*, 2018.

[117] Vignesh Ramanathan, Congcong Li, Jia Deng, Wei Han, Zhen Li, Kunlong Gu, Yang Song, Samy Bengio, Chuck Rossenberg, and Fei Fei Li. Learning semantic relationships for better action retrieval in images. In *CVPR*, 2015. ISBN 9781467369640. doi: 10.1109/CVPR.2015.7298713.

[118] Joseph Redmon and Ali Farhadi. YOLO v.3. *Tech report*, 2018.

[119] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection. In *CVPR*, 2016. ISBN 978-1-4673-8851-1. doi: 10.1109/CVPR.2016.91.

[120] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *NIPS*, 2015.

[121] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why Should I Trust You?" Explaining the Predictions of Any Classifier. In *KDD*, 2016. ISBN 9781450342322. doi: 10.1145/2939672.2939778.

[122] Herbert Robbins and Sutton Monro. A Stochastic Approximation Method. *Annals of Mathematical Statistics*, 1951.

[123] Marcus Rohrbach, Michael Stark, György Szarvas, Iryna Gurevych, and Bernt Schiele. What helps where - and why? Semantic relatedness for knowledge transfer. In *CVPR*, 2010. ISBN 9781424469840. doi: 10.1109/CVPR.2010.5540121. URL `https://www.informatik.tu-darmstadt.de/fileadmin/user_upload/Group_UKP/publikationen/2010/RohrbachCVPR2010.pdf`.

[124] Alessio Sarullo and Tingting Mu. On Class Imbalance and Background Filtering in Visual Relationship Detection. 2019.

[125] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling Relational Data with Graph Convolutional Networks. 2017. URL `https://arxiv.org/pdf/1703.06103v3.pdfhttp://arxiv.org/abs/1703.06103`.

[126] Florian Schroff and James Philbin. FaceNet: A Unified Embedding for Face Recognition and Clustering. In *CVPR*, 2015.

[127] Matthew Schultz and Thorsten Joachims. Learning a Distance Metric from Relative Comparisons. In *NIPS*, 2003.

[128] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training Region-based Object Detectors with Online Hard Example Mining. In *CVPR*, 2016. URL `https://arxiv.org/pdf/1604.03540.pdf`.

[129] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. A general reinforcement learning algorithm that masters chess,

shogi, and Go through self-play. *Science*, 2018. ISSN 10959203. doi: 10.1126/science.aar6404.

[130] Edgar Simo-Serra, Eduard Trulls, Luis Ferraz, and Francesc Moreno-Noguer. FRACKING DEEP CONVOLUTIONAL IMAGE DESCRIPTORS. In *ICLR*, 2015.

[131] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR*, 2015. ISBN 9781450341448. doi: 10.1016/j.infsof.2008.09.005.

[132] Jake Snell, Kevin Swersky, and Twitter Richard Zemel. Prototypical Networks for Few-shot Learning. In *NIPS*, 2017.

[133] Richard Socher, Andrej Karpathy, Quoc V Le, Christopher D Manning, and Andrew Y Ng. Grounded Compositional Semantics for Finding and Describing Images with Sentences. *Transactions of the Association for Computational Linguistics*, 2014. URL `https://www.aclweb.org/anthology/Q14-1017`.

[134] Kihyuk Sohn. Improved Deep Metric Learning with Multi-class N-pair Loss Objective. In *NIPS*, 2016.

[135] Hyun Oh Song, Yu Xiang, Stefanie Jegelka Mit, and Silvio Savarese. Deep Metric Learning via Lifted Structured Feature Embedding. In *CVPR*, 2016.

[136] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild. Technical report, 2012.

[137] Robert Speer and Catherine Havasi. Representing General Relational Knowledge in ConceptNet 5. In *LREC*, 2012.

[138] Tanlin Sun, Bo Zhou, Luhua Lai, and Jianfeng Pei. Sequence-based prediction of protein protein interaction using a deep-learning algorithm. *BMC Bioinformatics*, 2017. ISSN 14712105. doi: 10.1186/s12859-017-1700-2.

[139] Muhammad Atif Tahir, Josef Kittler, Krystian Mikolajczyk, and Fei Yan. A multiple expert approach to the class imbalance problem using inverse random under sampling. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 5519 LNCS, pages 82–91, 2009. ISBN 3642023258. doi: 10.1007/978-3-642-02326-2{_}9.

[140] J R R Uijlings, K E A Van De Sande, T Gevers, and A W M Smeulders. Selective Search for Object Recognition. *International Journal of Computer Vision*, 2013.

[141] Laurens Van Der Maaten and Geoffrey Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9, 2008.

[142] Benjamin Van Durme and Lenhart Schubert. *Extracting implicit knowledge from text*. PhD thesis, 2010.

[143] Ashish Vaswani, Google Brain, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. In *NIPS*, 2017.

[144] Ivan Vendrov, Ryan Kiros, Sanja Fidler, and Raquel Urtasun. ORDER-EMBEDDINGS OF IMAGES AND LANGUAGE. In *ICLR*, 2016.

[145] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *CVPR*, 2015. doi: 10.1109/CVPR.2015.7298935.

[146] Oriol Vinyals, Google Deepmind, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching Networks for One Shot Learning. In *NIPS*, 2016.

[147] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and Tell: Lessons learned from the 2015 MSCOCO Image Captioning Challenge. *Tpami*, 99(PP):1–1, 2016. ISSN 0162-8828. doi: 10.1109/TPAMI.2016.2587640. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7505636`.

[148] Wenbin Wang, Ruiping Wang, Shiguang Shan, and Xilin Chen. Exploring Context and Visual Pattern of Relationship for Scene Graph Generation. In *CVPR*, 2019.

[149] Xiaolong Wang and Abhinav Gupta. Unsupervised Learning of Visual Representations using Videos. In *ICCV*, 2015. ISBN 1505.00687v2.

[150] Xiaolong Wang, Yufei Ye, and Abhinav Gupta. Zero-shot Recognition via Semantic Embeddings and Knowledge Graphs. In *CVPR*, 2018.

[151] Ziyu Wang, Nando de Freitas, and Marc Lanctot. Dueling Network Architectures for Deep Reinforcement Learning. *arXiv*, (9):1–16, 2016. ISSN 0163-6804. doi: 10.1109/MCOM.2016.7378425. URL `http://arxiv.org/abs/1511.06581`.

[152] Donglai Wei, Joseph Lim, Andrew Zisserman, and William T Freeman. Learning and Using the Arrow of Time. In *CVPR*, 2018.

[153] Kilian Q Weinberger, John Blitzer, and Lawrence K Saul. Distance Metric Learning for Large Margin Nearest Neighbor Classification. *Journal of Machine Learning Research*, 2009.

[154] Sanghyun Woo, Dahun Kim, Kaist Daejeon, Donghyeon EE Cho, and In EE So Kweon. LinkNet: Relational Embedding for Scene Graph. In *NIPS*, 2018. URL `https://arxiv.org/pdf/1811.06410.pdf`.

[155] Yongqin Xian, Christoph H. Lampert, Bernt Schiele, and Zeynep Akata. Zero-Shot Learning - A Comprehensive Evaluation of the Good, the Bad and the Ugly, 2018. ISSN 01628828.

[156] Eric P Xing, Andrew Y Ng, Michael I Jordan, and Stuart Russell. Distance Metric Learning, with Application to Clustering with Side-Information. In *NIPS*, 2002.

[157] Chenliang Xu, Shao Hang Hsieh, Caiming Xiong, and Jason J. Corso. Can humans fly? Action understanding with multiple classes of actors. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 07-12-June:2264–2273, 2015. ISSN 10636919. doi: 10.1109/CVPR.2015.7298839.

[158] Danfei Xu, Yuke Zhu, Christopher B. Choy, and Li Fei-Fei. Scene Graph Generation by Iterative Message Passing. In *CVPR*, 2017. doi: 10.1109/CVPR.2017.330.

[159] Kelvin Xu, Jimmy Lei Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S Zemel, and Yoshua Bengio. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *ICML*, 2015.

[160] Fei Yan and Krystian Mikolajczyk. Deep Correlation for Matching Images and Text. In *CVPR*, 2015.

[161] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph R-CNN for Scene Graph Generation. In *ECCV*, 2018.

[162] Bangpeng Yao, Xiaoye Jiang, Aditya Khosla, Andy Lai Lin, Leonidas Guibas, and Li Fei-Fei. Human action recognition by learning bases of action attributes and parts. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1331–1338, 2011.

[163] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A Gift from Knowledge Distillation: Fast Optimization, Network Minimization and Transfer Learning. In *CVPR*, 2017.

[164] Guojun Yin, Lu Sheng, Bin Liu, Nenghai Yu, Xiaogang Wang, Jing Shao, and Chen Change Loy. Zoom-Net: Mining Deep Feature Interactions for Visual Relationship Recognition. In *ECCV*, 2018.

[165] Ruichi Yu, Ang Li, Vlad I. Morariu, and Larry S. Davis. Visual Relationship Detection With Internal and External Linguistic Knowledge Distillation. In *ICCV*, 2017.

[166] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural Motifs: Scene Graph Parsing with Global Context. In *CVPR*, 2018.

[167] Hanwang Zhang, Zawlin Kyaw, Shih-Fu Chang, and Tat-Seng Chua. Visual Translation Embedding Network for Visual Relation Detection. 2017. doi: 10.1109/CVPR.2017.331. URL `http://arxiv.org/abs/1702.08319`.

[168] Ji Zhang, Mohamed Elhoseiny, Scott Cohen, Walter Chang, and Ahmed Elgammal. Relationship Proposal Networks. In *CVPR*, 2017.

[169] Ji Zhang, Yannis Kalantidis, Marcus Rohrbach, Manohar Paluri, Ahmed Elgammal, and Mohamed Elhoseiny. Large-Scale Visual Relationship Understanding. In *AAAI*, 2019.

[170] Ji Zhang, Kevin J Shih, Ahmed Elgammal, Andrew Tao, and Bryan Catanzaro. Graphical Contrastive Losses for Scene Graph Parsing. In *CVPR*, 2019.

[171] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful Image Colorization. In *ECCV*, 2016.

[172] Hang Zhao, Xavier Puig, Bolei Zhou, Sanja Fidler, and Antonio Torralba. Open Vocabulary Scene Parsing. In *ICCV*, 2017. URL `http://openaccess.thecvf.com/content_ICCV_2017/papers/Zhao_Open_Vocabulary_Scene_ICCV_2017_paper.pdf`.

[173] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. ISBN 9781538604571. doi: 10.1109/CVPR.2017.660.

[174] Yaohui Zhu, Shuqiang Jiang, and Xiangyang Li. Visual relationship detection with object spatial distribution. In *ICME*, 2017.

[175] Yuke Zhu, Alireza Fathi, and Li Fei-Fei. Reasoning about Object Affordances in a Knowledge Base Representation. In *ECCV*, 2014. URL `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.674.8037&rep=rep1&type=pdf`.