View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Savoirs UdeS

LA RECONNAISSANCE D'INTENTION PAR APPRENTISSAGE PROFOND À L'AIDE DE CONNAISSANCES SYMBOLIQUES

par

Mariane Maynard

Mémoire présenté au Département d'informatique
en vue de l'obtention du grade de maître ès sciences (M.Sc.)

FACULTÉ DES SCIENCES

UNIVERSITÉ DE SHERBROOKE

Sherbrooke, Québec, Canada, 22 novembre 2019

Le 21 novembre 2019

le jury a accepté le mémoire de Madame Mariane Maynard
dans sa version finale.

Membres du jury

Professeur Froduald Kabanza
Directeur de recherche
Département d'informatique

Professeur Pierre-Marc Jodoin
Membre interne
Département d'informatique

Professeur Dave Touchette
Président-rapporteur
Département d'informatique

Sommaire

Pouvoir inférer l'intention de personnes que l'on observe ou avec lesquelles on interagit, ou de personnages d'histoires qu'on lit ou l'on nous raconte est possiblement un des constituants les plus remarquables de l'intelligence humaine. Cette capacité cognitive, connue entre autres sous l'appellation *reconnaissance d'intention*, demeure pourtant un problème irrésolu en intelligence artificielle. Celle-ci profiterait grandement de cette habileté à travers de nombreuses applications, telles que des dialogueurs virtuels plus fluides, des véhicules autonomes qui anticipent mieux les mouvements des usagers de la route, et des maisons autonomes à l'écoute de leurs occupants.

L'apprentissage profond a récemment fait des percées éminentes en vision de l'ordinateur et en traitement du langage naturel. Il existe pourtant très peu d'applications au problème de reconnaissance d'intention, hormis à certains problèmes reliés comme la reconnaissance d'actions et d'activités, qui n'impliquent pas de longues séquences d'interaction planifiées pour atteindre un but. Une grande partie de la recherche de ce côté utilise des méthodes symboliques, qui sont basées essentiellement sur des connaissances d'experts humains. Or, ces méthodes sont incapables de s'adapter lorsque ces connaissances sont erronées, ce qui est un des freins majeurs à leur application sur des domaines réels.

Ce mémoire vise dans un premier temps à étudier le potentiel de l'apprentissage profond pour la reconnaissance d'intention de manière expérimentale en comparaison avec des méthodes basées sur les coûts qui font partie de l'état de l'art symbolique. Dans un deuxième temps, il présente une manière de permettre aux réseaux de neurones d'améliorer leur capacité de généralisation grâce à des caractéristiques générées par des planificateurs symboliques lui offrant une conception des futurs potentiels de l'agent observé. Cela sera fait par l'introduction de deux articles scientifiques, dont le premier a été publié à PAIR, un événement concomitant à AAAI reconnu pour ses recherches sur la reconnaissance de plan, d'activités et d'intention, et dont le deuxième vient d'être soumis à AAAI, une conférence renommée en intelligence artificielle.

Mots-clés: Intelligence artificielle ; reconnaissance d'intention ; reconnaissance de plan ; apprentissage profond ; connaissances symboliques.

Remerciements

Je tiens d'abord à remercier mon directeur, Froduald Kabanza, qui m'a offert l'opportunité de continuer à la maîtrise sur un sujet aussi ambitieux. La patience et la persévérance dont il a fait preuve pour m'accompagner tout au long de ce chemin périlleux ont toute ma gratitude. J'aimerais également remercier Hugo Larochelle, qui m'a généreusement attribué de son temps et de ses conseils. Sa contribution a été beaucoup plus importante qu'il ne voudrait l'admettre.

Je souhaiterais spécialement remercier mon collègue Thibault Duhamel, qui est arrivé juste avant que je commence à me parler toute seule. Comme quoi deux paires d'yeux en valent mieux qu'une pour trouver ce que l'on cherche, surtout quand la première a fixé le même point depuis trop longtemps. De fait même, je remercie tous mes anciens et présents collègues et étudiants au laboratoire et ailleurs pour m'avoir tenu compagnie ne serait-ce qu'une session. Je savais que je pouvais compter sur Victor, Thierry, Fred et les autres lorsque le besoin pour une bière à 5 h se faisait ressentir, et sur Raph pour sa passion contagieuse pour StarCraft ; ce fut respectivement réconfortant et motivant.

Je désirerais enfin remercier ma famille qui m'a soutenue tout au long de mes études et qui m'a encouragée à persévérer dans les moments difficiles. Je remercie tout particulièrement mon conjoint Benjamin Rancourt, à qui je dédie ce mémoire. Ben, si tu n'avais pas été là, ces lignes et celles qui suivent ne se seraient jamais écrites.

Abréviations

AAAI *Association for the Advancement of Artificial Intelligence*

AI Intelligence artificielle (*Artificial Intelligence*)

BFS Recherche en largeur (*Breadth-First Search*)

CNN Réseau de neurones convolutif (*Convolutional Neural Network*)

DL Apprentissage profond (*Deep Learning*)

DNN Réseau de neurones profond (*Deep Neural Network*)

FC Complètement connecté (*Fully Connected*)

HMM Modèle de Markov caché (*Hidden Markov Model*)

LSTM Longue mémoire à court terme (*Long Short-Term Memory*)

PAIR *Plan, Activity and Intent Recognition*

PDDL *Planning Domain Description Language*

SBR *Symbolic Behavior Recognition*

STDNN Réseau de neurones profond spatiotemporel (*Spatiotemporal Deep Neural Network*)

Table des matières

Liste des figures

Liste des tableaux

Introduction

La reconnaissance d'intention est l'habileté à reconnaître l'intention d'une personne en observant son comportement, en écoutant ou en lisant une narration de celui-ci, ou en dialoguant avec elle. Les humains en sont naturellement dotés et la développent dès un très jeune âge. Warneken et Tomasello [57] ont montré que les bambins étaient naturellement enclin à l'altruisme, et qu'ils démontraient déjà la capacité de reconnaître ce qu'une personne tentait de faire – autrement dit, son intention – et qui plus est, si elle avait besoin d'aide et quel était le meilleur moyen pour l'assister. Il est plausible que les humains aient développé cette aptitude pour leur survie, puisqu'ils s'en servent pour collaborer sur des objectifs communs et ainsi profiter de la force du groupe [58]. D'autres animaux sociaux en seraient aussi pourvus, quoique certains s'en serviraient à d'autres fins comme tromper des rivaux ou anticiper le danger [52].

Il est donc clair que la reconnaissance d'intention est un concept ancré chez l'humain qui régit sa manière d'interagir avec l'autre. Plusieurs applications machine, comme les maisons intelligentes, les véhicules autonomes [6], les dialogueurs virtuels [8] et j'en passe, bénéficieraient d'être dotées de cette capabilité, puisqu'elle leur permettrait une interaction plus fluide et naturelle avec leurs usagers.

L'intention chez l'humain existe en fait selon plusieurs niveaux d'abstraction [33]. Le niveau le plus bas concerne les gestes moteurs, qui sont souvent exécutés inconsciemment. Le niveau intermédiaire concerne des actions et activités simples, telles que prendre ou déposer un objet, se lever, s'asseoir ; ou se brosser les dents, démarrer la cafetière, marcher vers la table, etc. Ces actions et activités arrivent de manière immédiate ou ont une durée limitée dans le temps, et l'intention qui les porte n'existe que pendant leur exécution. Le dernier niveau d'abstraction concerne quant à lui les objectifs à long terme. Il requiert de savoir planifier une suite d'activités, d'actions et de gestes pour y arriver. C'est pourquoi le problème de reconnaissance d'intention est relié de manière intrinsèque aux problèmes de reconnaissance de plan, de but et d'activités.

Schmidt *et al.* [43] définissent le problème de reconnaissance de plan comme étant celui de prédire les prochaines actions d'un agent par observation de son comportement, ainsi que d'inférer le but qui les explique. La reconnaissance de but, quant à elle, concerne uniquement le problème d'inférer le but. L'intention peut être définie comme étant un engagement à exécuter un certain plan pour atteindre un but. On prend comme postulat que le comportement de l'agent observé est orienté vers un objectif, c'est-à-dire que toutes les actions sont entreprises afin de l'atteindre. De ce fait, la reconnaissance de plan englobe la reconnaissance de but, et la reconnaissance d'intention réunit à la fois le problème de la reconnaissance de plan et celui de la reconnaissance d'activités. Dans ce mémoire, nous nous concentrerons exclusivement au problème d'inférence des buts, mais utiliserons aussi les termes « reconnaissance de plan » et « reconnaissance d'intention » de manière interchangeable.

Au fil des ans, plusieurs chercheurs en intelligence artificielle ont abordé ce problème sous plusieurs angles différents. Ils ont étudié les modèles de Markov cachés [5] et de manière plus générale, les réseaux bayésiens dynamiques [7], la logique markovienne [41], ainsi que les grammaires probabilistes [13]. Ces approches ont toutes en commun qu'elles sont symboliques et utilisent une bibliothèque de plans qui délimite comment l'agent observé agit. Elles sont de toute évidence inefficaces lorsque l'agent n'agit pas selon les règles de la bibliothèque de plans.

Durant la dernière décennie, une approche symbolique qui permet de s'affranchir de cette bibliothèque s'est développée : c'est la reconnaissance de but basée sur les coûts [39, 27]. Plutôt que de fournir une bibliothèque de plans, on fournit une description du domaine et du problème, appelée théorie du domaine [49], et l'on utilise un planificateur pour trouver des plans possibles de l'agent à la volée selon ses actions observées. L'approche s'appuie sur le principe de rationalité : on suppose que l'agent cherchera à prendre le plus court chemin – ou le moins coûteux – pour se rendre à destination au meilleur de ses connaissances. Baker *et al.* [3] ont en fait pu montrer que l'humain lui-même inférait toujours le but le plus en ligne directe avec les observations. Les approches basées sur les coûts utilisent donc les coûts des plans possibles générés par des planificateurs optimaux pour inférer le but de l'agent, selon l'idée que plus le coût du plan actuel s'éloigne du coût optimal pour un but, moins ce but devient vraisemblable.

Quoique plus flexibles que les approches à bibliothèque de plans, les méthodes basées sur les coûts sont aussi limitées par la présomption rigide qu'elles font sur la rationalité de l'agent. En effet, elles retournent des résultats contre-intuitifs lorsqu'il est un tant soit peu sous-optimal [29]. Bien que l'humain cherche à être rationnel, il ne l'est néanmoins pas complètement. Qui plus est, ces approches se basent sur plusieurs autres connaissances de l'environnement qui peuvent être aussi sujettes à erreurs.

C'est pourquoi il devient intéressant de se tourner vers l'apprentissage de représentations à partir de données brutes du comportement de l'agent dans son environnement. L'apprentissage profond a véritablement fait des avancées majeures dans cette direction, en particulier pour la vision par ordinateur et la compréhension du langage naturel. Ces avancées n'ont pas tardé à être exploitées pour la reconnaissance d'actions directement à partir de vidéos [46, 24], la reconnaissance d'activités [36, 56], ainsi que la prédiction de la direction prise des usagers de la route [6]. Par contre, ces approches ne sont pas concernées par les buts hauts niveaux poursuivis par l'agent. De ce côté, les avancées de l'apprentissage profond ne sont pas encore aussi tangibles [4, 32, 1, 35].

L'apprentissage profond peut aussi servir à reconnaître l'intention lors d'un dialogue [60, 8]. À ce moment, le sujet est conscient qu'on essaie de reconnaître son intention, surtout si le dialogueur n'est qu'une composante d'une application à son service, et l'on peut aisément déduire qu'il sera collaboratif. En effet, si le dialogueur lui demande ce qu'il souhaite faire, il risque de répondre franchement. Cependant, nous nous intéressons dans ce mémoire au problème de reconnaissance d'intention dans un contexte où l'utilisateur n'est pas conscient ou indifférent du fait qu'il est observé. Il ne fera donc rien pour aider ni contrer le processus de reconnaissance d'intention, du moins pas délibérément.

La perception visuelle et la compréhension du langage sont deux exemples de savoir-faire auxquels l'humain excelle sans pour autant être en mesure de bien l'expliquer. La reconnaissance de l'intention pourrait très bien tomber dans la même catégorie puisqu'elle survient souvent de manière inconsciente, raison pour laquelle il devient avantageux de s'intéresser à l'apprentissage profond. Pourtant, en vision par ordinateur comme en traitement du langage naturel, l'apprentissage n'y est pas pour tout : les méthodes de pointe de ces deux champs applicatifs ont grandement été aidées par certaines connaissances expertes qui ont guidé les choix architecturaux des réseaux de neurones qui les composent. Les réseaux de neurones à convolutions, par exemple, exploitent la proximité des pixels et la disposition en deux dimensions d'une image, tandis que les réseaux de neurones récurrents mémorisent un état interne utile au traitement de données séquentielles. Bien que la reconnaissance d'intention recoupe ces capacités à se situer dans l'espace-temps, elle en implique d'autres lorsqu'elle est portée par des buts à long terme, comme la capacité à planifier ou de manière plus générale, à *imaginer* le futur. Il est donc utile d'impliquer des connaissances symboliques spécifiques à la capabilité à reconnaître l'intention, de manière à orienter l'apprentissage.

Ce mémoire se divise en deux chapitres. Dans le premier, nous comparons la performance de l'apprentissage profond à reconnaître l'intention sur plusieurs problèmes référentiels à celle des approches symboliques basées sur les coûts. Nous explorons plusieurs architectures de réseaux de neurones de l'état de l'art telles que les réseaux de neurones convolutifs, les réseaux de neurones récurrents à longue mémoire à court terme, ainsi que de simples réseaux à couches entièrement connectées. Dans le deuxième, en nous concentrant cette fois-ci aux problèmes de navigation dans une grille, nous tentons d'améliorer la capacité des réseaux à généraliser leur performance à plusieurs environnements en le couplant avec les approches basées sur les coûts par l'intermédiaire d'une caractéristique d'entrée inventée spécifiquement pour les problèmes de reconnaissance d'intention. Cette caractéristique, intitulée *gradients de coûts*, est générée dynamiquement à l'aide de planificateurs optimaux. Enfin, ce mémoire conclut sur des suggestions pour des travaux futurs impliquant d'autres combinaisons possibles des connaissances symboliques à l'apprentissage profond pour la reconnaissance d'intention.

Chapitre 1

La reconnaissance de plan basée sur les coûts contre l'apprentissage profond

Résumé

La reconnaissance de plan demeure un défi de recherche de taille en IA depuis les années 70 [27]. Celle-ci s'est concentrée sur des approches symboliques basées sur des connaissances fournies explicitement par des experts, difficilement adaptables aux cas pratiques qui impliquent des observations brutes. Une des approches étudiées est celle basée sur les coûts [25, 18], qui s'appuie sur l'intuition que l'agent observé cherchera à minimiser le coût du plan vers le but qu'il poursuit, assumant qu'il est sensiblement rationnel.

D'un autre côté, l'apprentissage profond a fait des percées majeures en vision par ordinateur. Il commence à être exploité pour des problèmes d'analyse comportementale, comme la reconnaissance d'actions simples telles que marcher, parler, etc. directement à partir de vidéos [28, 16, 23]. Il devient donc intéressant de l'appliquer, par extension, à des problèmes de reconnaissance de plan

issus de la vie réelle. Pourtant, il reste encore peu étudié dans ce contexte, et la plupart des quelques méthodes existantes impliquent encore des connaissances d'experts [4, 13, 2, 22].

Nous présentons ici une analyse de la capacité d'architectures sélectionnées de réseaux de neurones profonds à inférer les buts en comparaison à des approches symboliques basées sur les coûts. Pour l'instant, nous avons utilisé cinq domaines référentiels issus de la littérature impliquant des observations synthétiques, afin d'évaluer les défis à relever pour son application éventuelle sur des cas pratiques. Plusieurs architectures familières ont été étudiées, notamment les réseaux de neurones à convolutions, les réseaux récurrents à longue mémoire à court terme, ainsi que les réseaux complètement connectés. Les résultats montrent que les réseaux infèrent le but avec une meilleure précision et plus rapidement que les algorithmes basés sur les coûts, mais ne peuvent pas généraliser à plusieurs environnements d'un même domaine.

Commentaires

Une première version de cet article a été publiée à l'événement *Plan, Activity and Intent Recognition* (PAIR) en 2019 [1] concomitant avec la conférence de l'*Assiociation for the Advancements of Artificial Intelligence* (AAAI). Ceci est une version améliorée. Elle sera soumise sur arXiv [2].

Ces travaux ont été réalisés par Mariane Maynard en collaboration avec Thibault Duhamel dans le cadre de sa maîtrise en informatique. Thibault s'est occupé de mettre en place la méthode expérimentale suggérée et présentée par Mariane. Le professeur Froduald Kabanza a supervisé la réalisation, validation et rédaction de l'article.

1. Actes archivés de PAIR 2019 : http://www.planrec.org/PAIR/Resource_files/PAIR19papers.zip
2. https://arxiv.org/

Cost-Based Plan Recognition Meets Deep Learning [3]

Mariane Maynard, Thibault Duhamel, Froduald Kabanza
Département d'informatique
Université de Sherbrooke
Sherbrooke, Québec (Canada) J1K 2R1
mariane.maynard@usherbrooke.ca,
thibault.duhamel@usherbrooke.ca,
froduald.kabanza@usherbrooke.ca

Abstract

The ability to observe the effects of actions performed by others and to infer their intent, most likely goals, or course of action, is known as a plan or intention recognition cognitive capability, and has long been one of the fundamental research challenges in AI. Deep learning has recently been making significant inroads on various pattern recognition problems, except for intention recognition. While extensive research on intention recognition capabilities have been done since the seventies, the problem still remains unsolved for most interesting cases in various areas, ranging from natural language understanding to human behavior understanding based on video feeds. This paper compares symbolic inverse planning, one of the most investigated approaches to plan recognition, to deep learning using CNN and LTSM neural network architectures, on five synthetic benchmarks often used in the literature. The results show that the deep learning approach achieves better goal-prediction accuracy and timeliness than the symbolic cost-based plan recognizer in these domains. Although preliminary, these results point to interesting future research avenues.

3. An earlier version of this paper was published to PAIR (AAAI 2019 workshop).

1.1 Introduction

The ability to infer the intention of others, also known as goal, plan or activity recognition, is central to human cognition and presents a wide range of application opportunities in many areas. Human behavior is often the result of conscious and unconscious cognitive planning processes [27, 3]. Therefore, to infer the intention of other people interacting with us, our brain is somehow able to predict what might be their goals or plans based on observations of clues from their actions. This capability is central to interact smoothly with people, to avoid danger in many situations, and to understand situations unfolding before us, such as predicting the behaviors of pedestrians when driving. Not surprisingly, there is intense research on intention recognition on many AI problems ranging from natural language understanding [36] and human-machine interaction [7] to autonomous vehicles [34] and security monitoring.

Intention recognition can be seen as part of the larger problem of pattern recognition, with the important nuance that it deals with goal-oriented patterns. Deep learning has been making significant inroads in recognizing patterns in general. Latest computer vision algorithms are now able to identify simple human behaviors involving short sequences of actions from videos, such as talking, drumming, skydiving, walking, and so on [28, 14, 38]. However, recognizing behaviors involving longer goal-oriented sequences of actions and produced by elaborate planning processes is another challenge yet barely tackled by end-to-end deep learning solutions [19, 20, 1].

For a long time, various symbolic inference paradigms have been experimented to try to infer the intention from observations based upon handcrafted models, using probabilistic inference frameworks such as HMM [5], Dynamic Bayesian Networks [6], Markov logic [26], probabilistic grammar parsing [11], cost-based plan recognition [25, 18], etc. These approaches require that human experts provide models of behaviors (e.g., domain theories or plan libraries [31]) that serve as input to the plan recognition inference engine. However, as vision, language understanding, and other perception tasks, intent recognition is a know-how difficult to express in a model, and this often results in a biased or utterly inaccurate definition of the domain for the inference engine. The appeal of representational learning is indeed the ability to extract modeling features, otherwise difficult to explain for an expert, from data.

In this paper, we show that familiar deep neural networks architectures, namely CNNs and LTSM networks, can perform well on intention recognition problems in navigation domains compared to a symbolic cost-based plan recognition algorithm considered as the state of the art on this problem [25, 18]. In intention recognition for navigation, we consider the case of an agent (the observer) inferring the goal of another agent (the observee) navigating in an environment, for which the map is known *a priori*, and for which there is a fixed number of points of interest which could be the potential destinations of the observee. This is an academic benchmark, with some simplifications. It should be understood as a step towards solutions that will work eventually in more realistic environments.

While preliminary, the results show that a CNN gives better and quicker goal-prediction accuracy than the state-of-the-art symbolic plan recognition method. Comparisons on other academic benchmarks often used to evaluate symbolic plan recognizers also suggest that deep neural networks offer competitive performance. It seems that even a simple fully connected network is able to learn abstractions underlying sequential decisions conveyed in the observed patterns of a goal-directed agent enough to outperform a cost-based approach. Before these experiments, we expected the latter to perform better since it is inherently tailored to deal with sequential decisions. These surprising results raise interesting avenues of investigation that we discuss in the paper.

The rest of the paper follows with a brief review of the most related work, background, experiment methodology, experiment results and a conclusion.

1.2 Related Work

A few approaches combine deep learning and symbolic inference in different ways. For example, Granada *et al.* [13] use a deep neural network to recognize individual actions of an actor cooking recipes in a kitchen, and then use a symbolic algorithm, SBR, to infer the goal underlying an observed sequence of actions. This approach requires as input the sequence of actions recognized by the neural network and a handcrafted model (plan library) representing abstractions of potential plans the observed agent could execute. There is no mechanism allowing the handcrafted plan library to adapt its own abstractions to classification errors of the neural network recognizing individual actions.

The procedure in Bisson *et al.* [4] is also based on a symbolic algorithm, which requires as input a sequence of observations of actions performed by an agent and a plan library. One component of the plan library representation is a probabilistic model of the choices the observed agent could make when selecting and executing plans from the plan library. A neural network learns this probabilistic model whereas the rest of the plan library is handcrafted.

In both approaches, goal inferences or plan predictions are done by a symbolic inference engine, not a deep neural network. Deep learning is involved only as an auxiliary procedure either to scan individual actions [13], or to learn a probabilistic model [4]. In contrast, in the experiments we discuss herein, a neural network does all the inference.

To the best of our knowledge, Min *et al.* [19] are among the first to use a plan recognition pipeline only made of a neural network. They use feed-forward n-gram models to learn the player's objective from a sequence of his actions in the Crystal Island game. The follow-up method in 2016 [20] uses Long Short-Term Memory (LSTM) networks, better suited to learn patterns in sequences. In both approaches, the features fed to the neural network were engineered instead of merely being raw player's events such as mouse clicks and key presses. While these methods demonstrate interesting results in a specific domain, they do not include a systematic comparison to symbolic ones.

Amado *et al.* [1] more recently introduced a deep learning pipeline to recognize the goal achieved by a player in different simple games (such as 8-puzzle, tower of Hanoi...) from raw images, divided into 3 steps. First, they convert inputs into a latent space (which is a representation of state features) using a previous auto-encoder algorithm [2]. The properties of the latent space are built to be reminiscent of a PDDL state representation. Then, an LSTM network utilizes it to perform a regression task, that is building a goal prediction in the latent space. Finally, the decoder reconstructs the goal image from its representation. While this approach does perform well on simple task-planning problems, it may not be applicable in real life settings. The method indeed tries to extract an approximate domain structure (states representation reminiscent of a PDDL) from temporal changes in observation sequences, and it is unsure whether or not real data can be exploited to frame such rules.

Thus, although deep learning has started to be investigated for plan recognition in different approaches, we are not aware of any systematic comparison using an end-to-end deep-learning pipeline for plan recognition versus using a symbolic or hybrid approach like those discussed above. In particular, we are not aware of any comparison between symbolic cost-based plan recognition and neural networks trained directly and only on raw observations, which is the experiment specifically discussed herein.

1.3 Background

In order to follow the methodology used for the experiments, it is useful to have some background on deep neural networks and cost-based plan recognition. Let us first recall the definition of a plan recognition problem.

1.3.1 The Problem

The plan recognition problem is to infer the goal pursued by an actor from an observed sequence of effects of his actions, and also to extract the plan pursued by the actor from these observations [27]. There is a link between goals, plans and intention. A plan is a sequence of actions achieving a goal, whereas an intention is a commitment to executing a plan. It could be argued that from a plan one can infer the likelihood of goals and vice-versa. Thus, in the AI literature, plan recognition has come to encompass all problems related to understanding goal-oriented behaviors, whether the focus is on inferring the goal, inferring intention, predicting the plan, or combinations of those three.

The experiments discussed herein deal with inferring the distribution probability of goals by observing action effects. Given a sequence of observations $o_\pi = o_1, \ldots, o_n$, – that may come directly from sensors or followed by relative prior parsing and processing – and a set G of potential goals that the agent might pursue, the problem is to infer a posterior probability distribution across G, $P(G|o_\pi)$, representing the probabilities that the agent might be pursuing a goal given the observations. Note that a plan recognition problem is also a pattern recognition problem, but not vice-versa. That is, not all pattern recognition algorithms are geared towards goal-directed behaviors, let alone, towards inferring the goals underlying goal-directed behaviors.

1.3.2 Deep Learning

It is easy to cast a plan recognition problem as a supervised deep-learning problem. In fact, the approach does not differ from other plan recognition problems.

Given a set of sequences of observations $\mathcal{O}$ and a set of potential goals G, let us assume that there exists a true recognition function f that maps perfectly each $o_\pi \in \mathcal{O}$ to its true goal $g_{o_\pi} \in G$, that is, $f(o_\pi) = g_{o_\pi}$.

While f is unknown (this is what we want to infer), we assume we have access to a training dataset of paired examples (o_π, g_{o_π}), i.e. we know the true goal g_{o_π} for some $o_\pi \in \mathcal{O}$. A supervised learning algorithm will seek to approximate f with a function f' parameterized by some set of parameters θ that minimizes the number of erred predictions in our dataset of examples. In other words, f' minimizes

$$L = \sum_{n=0}^{N} l(f'(o_\pi^n; \theta), g_{o_\pi^n})$$

where l is a loss function that is 0 when f' predicts accurately, and > 0 otherwise.

A single-layer neural network uses a simple linear transformation of the input using weight and bias parameters followed by a non-linear function in place of f':

$$f'(o_\pi) = \gamma(W o_\pi + b)$$

where W and b are the weight and bias parameters, respectively, and γ is a non-linear function such as sigmoid, hyperbolic tangent (tanh), rectifier linear units (ReLU), softmax, etc. A (deep) neural network is a composition of several of these transformations, usually with a different set of parameters at each layer [12]. These parameters are trained in order to converge to the minimum of the objective, usually by gradient descent of the loss function.

There exist specialized types of networks that process data differently and are more fit to some forms of input and problems. For instance, convolutional neural networks (CNNs) use filters of parameters and the convolution operation to process 2D input such as images or spatial information. Recurrent neural networks (RNNs) can memorize an internal state and process sequences of inputs, such as observed actions, making them better adapted to analyze dynamic behaviors than simple feed-forward networks are. Long Short-Term Memory networks (LSTM) used by Min *et al.* [20] are types of RNNs that allow for better gradient propagation and thus show better learning results than vanilla RNNs on longer sequences.

1.3.3 Symbolic Cost-Based Plan Recognition

The intuition behind cost-based plan recognition is the *principle of rationality*: people tend to act optimally to the best of their knowledge [3] and motor skills. Thus, one could infer the goal of an observed agent by trying to reason from the observed agent's point of view, that is, trying to invert his planning process. This does not mean we need to know his planning process per se.

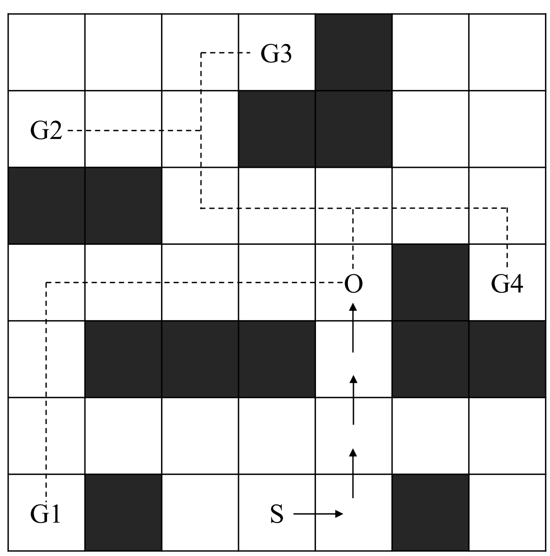

Figure 1.1 – A navigation grid example, where the agent is constrained with obstacles.

As noted by Ramírez and Geffner [25], given a sequence of observations, we could infer the probability that a given goal is the one being pursued by an agent by evaluating if his behavior observed so far is economical and might indeed commit to reaching that goal. To illustrate, consider the map in figure 1.1, representing areas of interest (goals) $G_1, \ldots, G_4$, obstacles, and a sequence of observations of an agent moving around, starting from position S. From the observation so far $o_\pi = o_1 \rightarrow \ldots \rightarrow o_4$, the agent logical goal is unlikely G_1, since we can find a shorter path from its start state to G_1 than the one it is currently taking. Intuitively, we can derive the likelihood of a goal by comparing the cost of an optimal plan to the goal consistent with the observations with the cost of an optimal plan to the goal regardless of the observations. The higher the difference between these two costs is, the less likely the goal is. Formally, the likelihood of an observation sequence $\mathcal{O}$ to reach a goal g can be inferred as:

$$P(o_\pi|g) = \frac{e^{-\beta\Delta(s,g,o_\pi)}}{1 + e^{-\beta\Delta(s,g,o_\pi)}}$$

where β is a positive constant determining how optimal we assess the observed agent's behavior to be. Δ is defined to be:

$$\Delta(s, g, o_\pi) = c(s, g, o_\pi) - c(s, g, \neg o_\pi)$$

where $c(s, g, o_\pi)$ is the cost of the optimal plan π_o between s and g that complies with the observations (i.e. all observed actions of o_π are embedded monotonically in the plan) and $c(s, g, \neg o_\pi)$ is the cost of the optimal plan $\pi_{\neg o}$ that does not comply with the observations (o_π is not embedded in π).

From $P(o_\pi|g)$, we can derive the posterior probability of the goal using the Bayes rule: $P(g|o_\pi) = \alpha P(o_\pi|g)P(g) \forall g \in G$, where $P(g)$ is the prior probability (often assumed to be uniform) and α is a normalization factor.

In principle, a planner can be used to compute plan costs [25]. However, computing a plan, even in the simple case of a deterministic environment under full observability, is NP-Complete [8]. This is not realistic in situations where an agent needs to infer the intention of others quickly. Approximate plan costs, computed by suboptimal planners which run faster than optimal planners can be used to infer approximate distribution [24]. This can be helpful in situations where the most important thing is to identify the most likely goals. Nonetheless, even heuristic planners which compute suboptimal plans still take too much time for most real-time applications.

We can avoid some calls to the planners by incorporating heuristic functions directly into the plan recognition inference process. Vered and Kaminka [33] introduced such heuristics that judge whether a new observation may change the ranking of goals or whether a goal can be pruned. However, they become useless in more complex problems where the goals cannot be pruned early and thus the number of calls to the planner is not reduced.

A practical approach to cost-based plan recognition is to compute the plan costs offline. This way, instead of invoking a planner, we have a lookup in a table or a map of plan costs. For navigation problems, where the issue is to predict the destination of an agent moving around, Masters and Sardiña [18] describe an approach for accurately pre-computing plan costs by relaxing Ramírez and Geffner [25]'s algorithm with – practically – no loss in accuracy. It is overall the same, but they compute the cost difference to instead be $\Delta(s, g, n) = c(n, g) - c(s, g)$ where n corresponds to the last seen position of the observed agent. This relaxation not depending on the whole observation sequence avoid computing as many different costs as needed by Ramírez and Geffner [25], making them easier to be stored beforehand. However, it is quite limited in application to the – discrete – navigation domain.

In general, however, there is no well-known method of accurately pre-computing and storing plan costs for all possible combinations of initial and goal states for an arbitrary domain. Sohrabi *et al.* [29] compute the top-k plans for each goal, and calculates the goal inference by summing the probability of plans in the set achieving this goal, where a likelihood of a plan is not only dependent of its cost but also to what degree it complies to the observations. The problem is that the required number of plans is high (1000) to have results comparable to Ramírez and Geffner [25]'s. Other various recent studies present different ideas to reduce planners' compute time. For instance, E.-Martín *et al.* [9] compute cost interaction estimates in plan graphs, while Pereira *et al.* [21] use landmarks, with the idea that goals with a higher completion ratio are more likely. However, their solutions are less accurate, since they are mere approximations of plans generated by an optimal planner.

1.4 Comparison Methodology

To compare cost-based plan recognition to deep learning, we used five synthetic domains often selected to evaluate the performance of a symbolic plan recognizer as referenced above. Ultimately, we want to evaluate plan recognizers using real-world benchmarks. Meanwhile, the synthetic domains can provide some useful insight.

1. Navigation: Predicting the goal destination of an agent navigating a map [17]. The domain consists of 20 maps from StarCraft, provided by MovingAI [4], down-scaled to 64x64 pixels, where the agent can perform actions limited to the first 4 cardinal directions. Plan recognition problems were generated by placing one initial position and 5 goals on the maps.
2. Intrusion Detection: Predicting the goals of network hackers with their activities [10]. The observed agent is a user who may perform attacks on 10 hosts. There are 6 possible goals that the hacker might reach by performing 9 actions on those servers. Observation sequences are typically between 8 and 14 observations long.
3. Kitchen: Inferring the activity of a cook in a smart home kitchen [37]. The cook can either prepare breakfast, lunch or dinner (possible goals) [37]. He can take objects, use them, and perform numerous high-level activities. Observation sequences are typically between 3 and 8 actions long.
4. BlocksWorld: Predicting the goal of an agent assembling 8 blocks labeled with letters, arranged randomly at the beginning [24]. Achieving a goal consists in ordering blocks into a single tower to spell one of the 21 possible words by the use of 4 actions. Observation sequences are typically between 6 and 10 actions long.
5. Logistics: Predicting package delivery in a transport domain. Six packages must be conveyed between 6 locations in 2 different cities, using 1 airplane, 2 airports and 2 trucks [24]. There are 6 possible actions available to achieve around 10 possible goals. Observation sequences are typically between 16 and 22 actions long.

4. MovingAI Lab: https://movingai.com/

The observation data for the 4 last benchmarks were fetched on https://github.com/pucrs-automated-planning/goal-plan-recognition-dataset.

For the navigation benchmark, we used four different neural network architectures (see figure 1.2): a fully connected network (FC), an LSTM network and two convolutional neural networks (CNN). We felt both the LSTM and CNN appropriate for this domain, given that the former usually performs well learning from sequences, whereas the latter is appropriate for learning from spatial data (maps in our case).

The first 3 networks were trained on problems generated from a single map. We additionally trained a convolutional network (CNNMultimaps) on multiple maps, regardless of their goals, start and obstacle positions, to see if and how it could generalize across multiple plan recognition navigation domains.

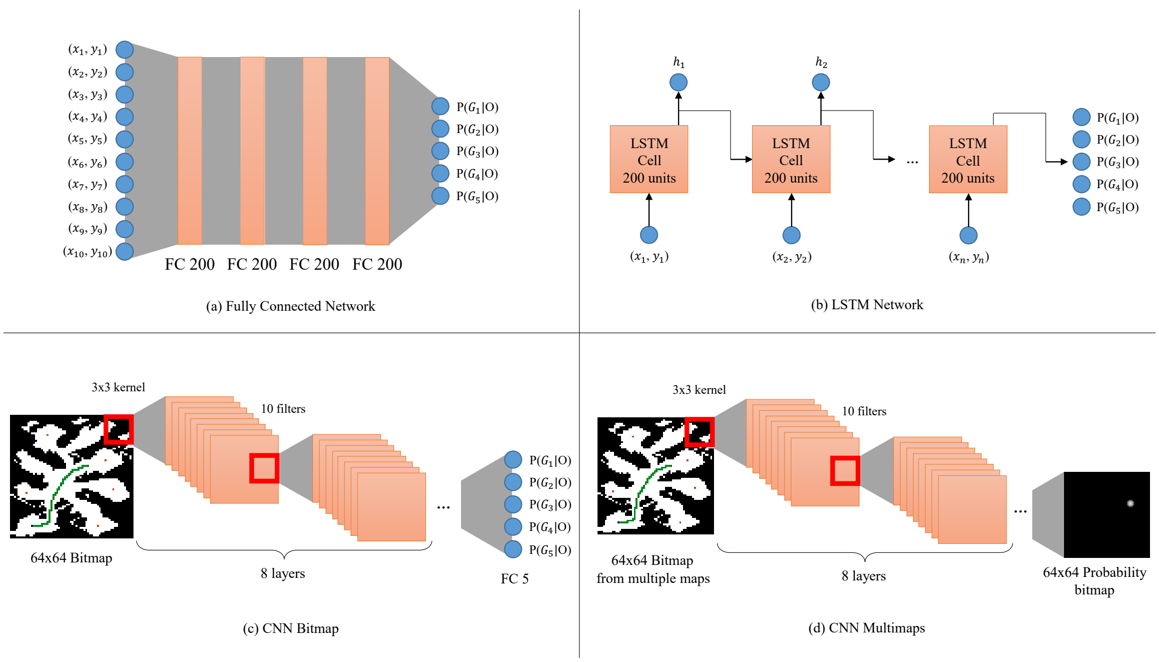

Figure 1.2 – Representation of our architectures for the navigation domain. (x_i, y_i) stands for the coordinates of the agent's location in the grid. (a), (b), and (c) were trained on a single map, while (d) was trained on multiple maps.

Here is a thorough description of the network architectures:

1. FC: this network is comprised of 4 dense layers of 200 units and one output layer of 5 units representing the goal probability distribution.

2. LSTM: this network as a single LSTM layer of 200 units and a dense output layer of 5 units.
3. first CNN (CNNBitmap): this network has 8 convolutional layers of 10 filters of size 3x3 respectively. The resulting features are flattened and passed to a dense layer of 5 units.
4. CNNMultimaps: the first 8 layers of this network are the same as the CNNBitmap's, but they are followed by an additional convolution layer of one 3x3 filter instead of a dense layer.

Since methods FC, LSTM and CNNBitmap were trained and tested on the same map, where goals were known in advance, it was possible to deduce a probability distribution array of fixed size (here 5). However, we could not make this assumption for the general fully convolutional method (CNNMultimaps) trained on multiple, different maps. This is why the latter instead outputs a probability distribution over the entire map, representing the belief that the agent's goal is at one position or another, allowing any number of goals and positions in general.

For the four other domains, we used a fully connected network with three dense layers of 256, 32 and 5 units respectively. We compare it with original Ramírez and Geffner [25]'s method, since there is yet no proven method for pre-computing plan costs – or approximations of them – for these domains without a significant loss in accuracy [9, 21, 33].

Besides the architecture, an implementation of a neural network involves the choice of specific parameters, activation functions, and optimization algorithm. Given that we want to find a correct goal amongst a set of possible ones and work with probabilistic scores, we quantify the loss with the categorical cross-entropy function and work with the accuracy metric, which is computed as the percentage of correct predictions. A prediction is said to be correct if its highest output probability corresponds to the true goal. In case of ties, we consider a random uniform draw between all the goals having the same top probability. In cost-based plan recognition literature, alternative accuracy metrics are often used, such as metrics using a threshold [21, 29], or simply an accuracy metric where ties are not randomly disambiguated and instead considered as an accurate prediction [25, 9, 29]. However, we find them highly artificial and unfit to evaluations of real-world applications, so we chose to only consider the top

1, which should account for lower accuracy values. It is also important to note that we apply the same metric to every method.

Hidden layers are activated with the ReLU function, while the output layer is activated with the softmax function. To train the networks, the Adam optimizer [15] is used, with a learning rate of 0.001, $\beta 1$ of 0.9, $\beta 2$ of 0.999 and no decay. To prevent overfitting, we also used dropout [30] for all layers with a drop chance set to 0.1 or 0.2. Finally, inputs were shuffled uniformly prior to training.

1.5 Experiments and Results

We present hereby the full experiments and discuss their results, including full details on how training and test data were generated. For all domains, the datasets are split 80%-20% for training and test.

1.5.1 Navigation Domain

As mentioned above, four networks were trained for the navigation benchmark. The first three (FC, LSTM, CNNBitmap) were trained for 15 epochs on observations from a single map, with 100 observed paths. CNNMultimaps was trained on all available observations of all maps for 100 epochs. To mimic suboptimal behavior, we started by generating noisy optimal paths to these goals with a modified A* algorithm.

As paths were generated, we truncated them to measure how our networks could handle early predictions in an online application: both training and test sets consist of partial or complete sequences of observations truncated at the first 25%, 50%, 75% and 100% of the sequence, such that we can evaluate performances for partial as well as complete observability. It is important to note that this notion of partial observability differs from the usual literature definition: in many papers [25, 21, 9, 29], a certain percentage of observations is missing, but across the *whole* sequence. In opposition to that, in order to mimic real-time predictions, we cut the observation sequences to a given percentage and every following observation is dropped. We estimate that this idea of early observability is more realistic as it enables online resolution of plan recognition problems.

We used (x, y) coordinates as input for the FC network and LSTM methods. As paths lengths may differ, we eventually retained a fixed number of positions among the ones available to form inputs of fixed size, padding with zeros shorter sequences. We input the map converted to a 4-channel image to both CNNs, where each channel is a bitmap displaying information about the initial position, the positions of the potential goals, the visited positions, and walkable positions that are neither of the above.

For Masters and Sardiña [17]'s method (labeled M-S), only the last position of sub-paths was used. Cost maps were generated using optimal paths returned by the A* algorithm and stored offline. To compute the posterior probabilities, we assumed prior probabilities to be uniform and used a value of 1 for the β parameter.

We compared the accuracy of those 4 different networks on test sets with M-S. Results are shown in figure 1.3. The Y-axis represents the average accuracy on 10 different maps. The X-axis refers to the percentage sampled from total paths in the test set.

As it can be seen, method CNNBitmap ranks first. The reason could be that the convolution filters of the network help reason about the 2D structure of the grid and the observed path, as expected. FC and LSTM methods perform well too, but it seems that learning from coordinates is more complicated, or more imprecise, that learning directly from bitmaps in such a navigation domain.

Surprisingly, M-S was outperformed at least by CNNBitmap and FC. The reason might be that generated A* tracks stayed somehow deterministic despite the noisy behavior and thus, even in the case where multiple optimal paths to a goal exist, similar paths were always chosen for that goal. The neural networks thus quickly learned to fit these specific paths, even though at first earlier subsets could possibly go to either goal. This bias in the data incorporated by the generation process could be seen as problematic, but we argue otherwise. In real-world applications involving human agents, people usually take the same road even when multiple ones that are as good – or even better – exist. Data is therefore not uniformly distributed between every candidate roads. The capacity of neural networks to learn this bias and adjust for certain context and individuals is one of the properties that makes them appropriate for plan recognition in real-life applications. Additionally, in the

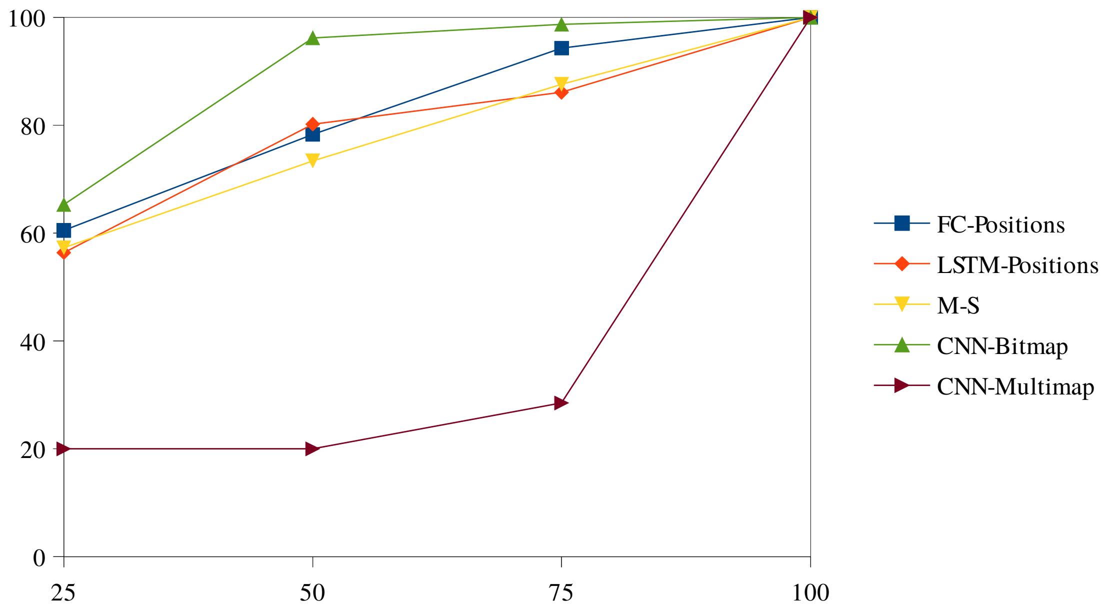

Figure 1.3 – Results of accuracy depending on the percentage retained from the complete observed path, in the navigation domains.

case of cost-based algorithms, even though all available data is used to compute costs, the final prediction is only achieved based on them, which represents a gradual loss of information.

The convolutional network trained and tested on all maps (CNNMultimaps) shows relatively bad early predictions (20% accuracy for 5 goals is just a random prediction), proving there is still room for improvement in order for neural networks to generalize to multiple maps. Nonetheless, the method can already create a link between a complete path and a goal (that is, learning but not predicting), and may be significantly improved by the use of specialized architectures, such as value iteration network [32] and visual relational reasoning [35]. We are currently working on improving its results.

Computing plan costs takes time, even offline. The results suggest that training neural networks, even if computationally complex, may be advantageous in this regard thanks to the trivially parallelizable nature of its operations and the computation power of modern hardware. However, a computation time comparison does not enlighten new advantages for this kind of context. Table 1.1 gives a summary of offline and online computation times. The LSTM networks have longer training times but may generalize better to longer sequences of observations with bigger sliding windows (since we fixed the maximum number of observations input to 10 and thus do not benefit fully from LSTM's training power over sequences). The CNN trained on multiple maps takes a long time to train but could have the potential to generalize to every navigation problem, so no additional training would be required for an unseen map. Symbolic approaches have no need of training nor dataset, but knowledge about the domain is required to handcraft the model and costs must be generated for every new map, whether it is offline or online (during prediction).

	T	P
FC	10 s	10μs
LSTM	30 s	4 ms
CNNBitmap	10 s	4 ms
CNNMultimaps	20 min	4 ms
R-G	0	1 s
M-S	7 s	10μs

Table 1.1 – Comparison of rough average computation times of the evaluated approaches on the navigation domain. T is the offline computation time, while P is the online prediction time.

1.5.2 Other Domains

The navigation benchmark deals with path-planning problems requiring much less knowledge than the other four domains. Those last benchmarks correspond to task-planning problems, involving constraints that differ from those in the navigation benchmark, thus requiring different kinds of domain representations. In fact, we represented them using the Planning Domain Definition Language (PDDL) as in Ramírez and Geffner [25].

A fully connected network was trained on each domain during 15 epochs, with a number of training examples ranging from 1000 to 3000 depending on the domain. We also trained an LSTM on these examples, but it ended up taking more time without providing significant result improvements.

A training example in the datasets is a sequence of observations from PDDL files, encoded with zeros or ones. Each observation in the sequence is one action type plus its arguments which are transformed to a one-hot vector. The neural network receives the complete sequence of transformed observations. To match a fixed input size, sequences shorter than the maximum size are padded with zeros and shifted $maxSize - size + 1$ times (for instance, if one observation is AB and the maximum size is 4, 3 new observations will be created: $AB00$, $0AB0$, $00AB$), hence generating new training data.

In the case of Ramírez and Geffner [25]'s method, labeled R-G, the costs were generated online, as first implemented by the authors, from optimal plans found by the HSP planner. The β parameter value was 1 and the prior probabilities of the goals were presumed to be uniform.

Results in figure 1.4 show the accuracy for both methods. The fully connected network outperforms the R-G approach almost every time. A similar explanation as for the navigation domain can be given for these results: generated sequences tend to be biased for each goal and the network learned it. In addition to providing higher prediction rates, networks are also quicker: on such plan recognition problems, the training part takes approximately 1 minute to infer reusable weights, while one prediction is made under 1ms. The R-G approach does not require training nor offline computation, but provides a prediction in minutes, sometimes hours, which is really long and cannot be applied to real-time decision making. A suboptimal planner might reduce computation times, but we can reasonably assume that it would still remain above several minutes or so for each goal prediction.

1.6 Conclusion

Although still preliminary, these results suggest that deep learning outperforms symbolic inverse planning, at least in the five domains considered. We plan to pursue this experimentation in real-world domains where we can gather data, including video games. We also plan to try different deep neural networks [12], symbolic plan recognition methods, multi-agent configurations for plan recognition, sensor limitations

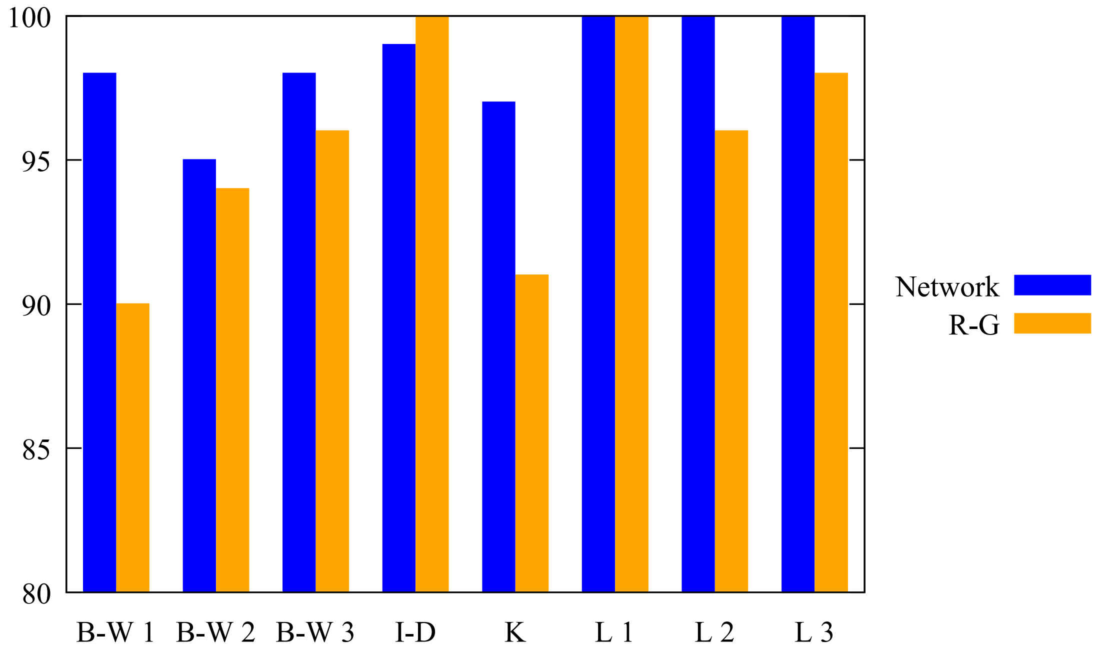

Figure 1.4 – Results of accuracy for the task-planning domains (B-W, I-D, K, and L stand for Blocks World, Intrusion Detection, Kitchen and Logistics respectively).

(partial observability vs full observability), attitudes between the observed agent and the observer (cooperative, adversarial, neutral) and different domains of application.

In some applications, it is important that the plan recognizer explains the rationale of its inferences. To do so, extracting a meaningful explanation from a neural network still remains a challenge. In contrast, the symbolic representation of symbolic plan recognizers makes the explanations easier, except that, as we have argued, those approaches are difficult to ground in real-world environments. This suggests that the exploration of hybrid approaches, such as those discussed in the related section, remains worth pursuing.

1.7 Acknowledgements

This work was supported by grants from the Natural Sciences and Engineering Research Council (NSERC) of Canada and the *Fonds de recherche du Québec – Nature et technologies* (FRQNT), and also by Compute Canada for the computing resources they provided. We are also thankful to Julien Filion, Simon Chamberland and anonymous reviewers for their insightful feedback that helped improve the paper.

References

[1] L. Amado, J. P. Aires, R. F. Pereira, M. C. Magnaguagno, R. Granada, F. Meneguzzi, "LSTM-Based Goal Recognition in Latent Space," *CoRR*, vol. abs/1808.05249, 2018. [Online]. Available: http://arxiv.org/abs/1808.05249

[2] M. Asai A. Fukunaga, "Classical Planning in Deep Latent Space: Bridging the Subsymbolic-Symbolic Boundary," in *AAAI 2018*, 2018. [Online]. Available: https://aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16302

[3] C. L. Baker, R. Saxe, J. B. Tenenbaum, "Action understanding as inverse planning," *Cognition 2009*, vol. 113, no. 3, pp. 329–349, 2009.

[4] F. Bisson, H. Larochelle, F. Kabanza, "Using a Recursive Neural Network to Learn an Agent's Decision Model for Plan Recognition," in *IJCAI 2015*, 2015, pp. 918–924.

[5] H. H. Bui, S. Venkatesh, G. A. W. West, "Policy Recognition in the Abstract Hidden Markov Model," *JAIR*, vol. 17, pp. 451–499, 2002.

[6] E. Charniak R. P. Goldman, "A Bayesian Model of Plan Recognition," *Artif. Intell.*, vol. 64, no. 1, pp. 53–79, 1993. [Online]. Available: https://doi.org/10.1016/0004-3702(93)90060-O

[7] C. Chen, X. Zhang, S. Ju, C. Fu, C. Tang, J. Zhou, X. Li, "AntProphet: an Intention Mining System behind Alipay's Intelligent Customer Service Bot," in *IJCAI 2019*, 08 2019, pp. 6497–6499.

REFERENCES

[8] G. F. Cooper, “The Computational Complexity of Probabilistic Inference Using Bayesian Belief Networks (Research Note),” *Artificial Intelligence*, vol. 42, no. 2-3, pp. 393–405, March 1990. [Online]. Available: http://dx.doi.org/10.1016/0004-3702(90)90060-D

[9] Y. E.-Martín, M. D. R.-Moreno, D. E. Smith, “A Fast Goal Recognition Technique Based on Interaction Estimates,” in *IJCAI 2015*, 2015, pp. 761–768.

[10] C. W. Geib R. P. Goldman, “Requirements for Plan Recognition in Network Security Systems,” in *International Symposium on Recent Advances in Intrusion Detection 2002*, 2002.

[11] C. W. Geib R. P. Goldman, “A probabilistic plan recognition algorithm based on plan tree grammars,” *Artificial Intelligence*, vol. 173, no. 11, pp. 1101–1132, 2009.

[12] I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning*. MIT Press, 2016.

[13] R. Granada, R. Pereira, J. Monteiro, R. Barros, D. Ruiz, F. Meneguzzi, “Hybrid Activity and Plan Recognition for Video Streams,” in *AAAI 2017*, 2017.

[14] J. Hou, X. Wu, J. Chen, J. Luo, Y. Jia, “Unsupervised Deep Learning of Mid-Level Video Representation for Action Recognition,” in *AAAI 2018*, 2018, pp. 6910–6917.

[15] D. P. Kingma J. Ba, “Adam: A Method for Stochastic Optimization,” *CoRR*, vol. abs/1412.6980, 2014.

[16] Y. Kong Y. Fu, “Human Action Recognition and Prediction: A Survey,” *CoRR*, vol. abs/1806.11230, 2018. [Online]. Available: http://arxiv.org/abs/1806.11230

[17] P. Masters S. Sardiña, “Cost-Based Goal Recognition for Path-Planning,” in *AAMAS 2017*, 2017, pp. 750–758.

[18] P. Masters S. Sardiña, “Cost-Based Goal Recognition in Navigational Domains,” *JAIR*, vol. 64, pp. 197–242, 2019. [Online]. Available: https://doi.org/10.1613/jair.1.11343

REFERENCES

[19] W. Min, E. Ha, J. P. Rowe, B. W. Mott, J. C. Lester, "Deep Learning-Based Goal Recognition in Open-Ended Digital Games," in *AIIDE 2014*, 2014.

[20] W. Min, B. W. Mott, J. P. Rowe, B. Liu, J. C. Lester, "Player Goal Recognition in Open-World Digital Games with Long Short-Term Memory Networks," in *IJCAI 2016*, 2016, pp. 2590–2596.

[21] R. F. Pereira, N. Oren, F. Meneguzzi, "Landmark-Based Heuristics for Goal Recognition," in *AAAI 2017*, 2017, pp. 3622–3628.

[22] R. F. Pereira, M. Vered, F. Meneguzzi, M. Ramírez, "Online Probabilistic Goal Recognition over Nominal Models," in *IJCAI 2019*, 2019, pp. 5547–5553. [Online]. Available: https://doi.org/10.24963/ijcai.2019/770

[23] H. Qian, S. J. Pan, B. Da, C. Miao, "A Novel Distribution-Embedded Neural Network for Sensor-Based Activity Recognition," in *IJCAI 2019*, 2019, pp. 5614–5620. [Online]. Available: https://doi.org/10.24963/ijcai.2019/779

[24] M. Ramírez H. Geffner, "Plan Recognition as Planning," in *IJCAI 2009*, 2009, pp. 1778–1783.

[25] M. Ramírez H. Geffner, "Probabilistic Plan Recognition Using Off-the-Shelf Classical Planners," in *AAAI 2010*, 2010.

[26] A. Sadilek H. A. Kautz, "Recognizing Multi-Agent Activities from GPS Data," in *AAAI 2010*, 2010. [Online]. Available: http://www.aaai.org/ocs/index.php/AAAI/AAAI10/paper/view/1603

[27] C. F. Schmidt, N. S. Sridharan, J. L. Goodson, "The Plan Recognition Problem: An Intersection of Psychology and Artificial Intelligence," *Artificial Intelligence*, vol. 11, no. 1-2, pp. 45–83, 1978.

[28] K. Simonyan A. Zisserman, "Two-Stream Convolutional Networks for Action Recognition in Videos," in *NIPS 2014* Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, K. Q. Weinberger, editors, 2014, pp. 568–576. [Online]. Available: http://papers.nips.cc/paper/5353-two-stream-convolutional-networks-for-action-recognition-in-videos.pdf

REFERENCES

[29] S. Sohrabi, A. V. Riabov, O. Udrea, "Plan Recognition As Planning Revisited," in *IJCAI 2016*, ser. IJCAI'16. AAAI Press, 2016, pp. 3258–3264. [Online]. Available: http://dl.acm.org/citation.cfm?id=3061053.3061077

[30] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[31] G. Sukthankar, C. Geib, H. H. Bui, D. Pynadath, R. P. Goldman, *Plan, Activity, and Intent Recognition: Theory and Practice*, 1st edition. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2014.

[32] A. Tamar, Y. Wu, G. Thomas, S. Levine, P. Abbeel, "Value Iteration Networks," in *IJCAI 2017*, 2017, pp. 4949–4953.

[33] M. Vered G. A. Kaminka, "Heuristic Online Goal Recognition in Continuous Domains," in *IJCAI 2017*, 2017, pp. 4447–4454.

[34] B. Volz, K. Behrendt, H. Mielenz, I. Gilitschenski, R. Siegwart, J. Nieto, "A data-driven approach for pedestrian intention estimation," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, Nov 2016, pp. 2607–2612.

[35] N. Watters, A. Tacchetti, T. Weber, R. Pascanu, P. Battaglia, D. Zoran, "Visual Interaction Networks," *CoRR*, vol. abs/1706.01433, 2017.

[36] T. Wen, Y. Miao, P. Blunsom, S. J. Young, "Latent Intention Dialogue Models," in *ICML 2017*, 2017, pp. 3732–3741. [Online]. Available: http://proceedings.mlr.press/v70/wen17a.html

[37] J. Wu, A. Osuntogun, T. Choudhury, M. Philipose, J. M. Rehg, "A Scalable Approach to Activity Recognition based on Object Use," in *ICCV 2007*, 2007.

[38] S. Yan, Y. Teng, J. S. Smith, B. Zhang, "Driver behavior recognition based on deep convolutional neural networks," in *ICNC-FSKD 2016*, 2016, pp. 636–641.

Chapitre 2

Fournir une idée du futur à une approche d'apprentissage profond pour la reconnaissance d'intention

Résumé

Bien que l'apprentissage profond continue de progresser dans la déduction d'intention dans les actions [13], les activités simples [20] et le dialogue [33, 6], très peu de progrès en comparaison ont été faits pour la reconnaissance d'intention à long terme. En effet, les approches actuelles ne peuvent pas généraliser à plusieurs environnements d'un même domaine [18, 1]. Notre hypothèse ici est que les architectures étudiées – notamment les réseaux récurrents à longue mémoire à court terme – bien que capables de traiter des données temporelles, sont dépourvues de la capacité à projeter l'agent dans des futurs possibles.

Les humains ont les connaissances et l'expertise nécessaires pour faire de la reconnaissance d'intention à court comme à long terme, car ils sont naturellement pourvus de cette capacité [31]. Par contre, d'une part par le fait que c'est un processus inconscient, ils ont de

la difficulté à exprimer ce savoir-faire en un modèle assez précis et complet pour pouvoir faire de l'inférence symbolique efficace.

Nous introduisons ici une méthode d'apprentissage profond exploitant les connaissances symboliques par la génération de caractéristiques d'entrée offrant une idée du futur grâce à des planificateurs optimaux. L'idée initiale se base sur les algorithmes de reconnaissance de plan basée sur les coûts [23, 15], et nous offrons dans un premier temps des coûts en entrée à un réseau de neurones profond spatiotemporel concaténés aux données d'observations. Dans un deuxième temps, nous décrivons une nouvelle caractéristique conçue spécialement pour le problème de reconnaissance d'intention, le *gradient de coûts*, qui permet au réseau de faire des inférences à la fois plus flexibles et perspicaces puisqu'il est plus riche en information.

Nous montrons que le réseau augmenté de cette caractéristique performe mieux qu'un réseau en étant dépourvu dans des domaines de navigation, et reste compétitif avec les approches purement basées sur les coûts. Nous montrons aussi que, grâce au pouvoir adaptatif de l'apprentissage profond à partir des données, notre solution est plus robuste aux erreurs de définition du domaine de planification qui aboutit en des comportements sous-optimaux ou d'apparence sous-optimale de la part de l'agent, malgré le fait que les gradients de coûts soient générés par des algorithmes symboliques.

Commentaires

Cet article a été soumis à la conférence de l'*Assiociation for the Advancements of Artificial Intelligence* (AAAI) de 2020 pour révision par les pairs. Ces travaux ont été réalisés et rédigés par Mariane Maynard sous la supervision du Professeur Froduald Kabanza, avec l'aide des commentaires de Thibault Duhamel.

Providing Future Insight to a Deep Learning Approach to Intent Recognition

Mariane Maynard, Thibault Duhamel, Froduald Kabanza
Département d'informatique
Université de Sherbrooke
Sherbrooke, Québec (Canada) J1K 2R1
mariane.maynard@usherbrooke.ca,
thibault.duhamel@usherbrooke.ca,
froduald.kabanza@usherbrooke.ca

Abstract

Being able to infer the intention of people we observe, interact with, or read stories about is one of the key hallmarks of human intelligence. While deep learning has been demonstrating significant progress in classifying human intent in natural language, dialogue, and basic activities, much less progress has been made on inferring the intent of agents engaged in long-term goal-directed behaviors. We introduce a new approach for inferring the intentions underlying such behaviors based on a novel idea on providing future insight to a spatiotemporal deep neural network. The gist of the idea is that, given an incremental sequence of observations of an agent and a symbolic model of the domain it evolves in, it is possible to generate insight about the agent's plausible futures facilitating intention inference learning in that domain. Unlike symbolic approaches to intent recognition, our approach is more robust to a possibly incomplete, erroneous or imprecise model. The approach demonstrates improved performance compared to a baseline deep learning devoid of the ability to generate future insight.

2.1 Introduction

The ability to infer the intention of other people is a hallmark of human intelligence, and it has been at the center of AI research for decades [25, 3, 30]. We infer the intention of others quite regularly, often unconsciously, when interacting, cooperating or competing with other people, or simply when passively observing their behaviors, reading or hearing to narrations about them.

An intention, in general, can be defined as the commitment to execute an action or a sequence of actions to achieve a specific and deliberate goal [25]. A goal-oriented sequence of actions also corresponds to a plan, hence inferring the intention of others is also related to plan, goal, and activity recognition problems. Herein, the focus is on inferring the likely goals of an agent from observations of his actions.

Classical approaches for intent recognition have relied on symbolic knowledge provided by human experts [30]. The main roadblock for these approaches is that, while humans are good at inferring intents [31], explaining the reasoning process underlying them precisely enough to produce effective symbolic knowledge is not as obvious since it is done unconsciously.

There is an increasing shift towards using deep learning to represent models of intent recognition from data, to avoid the drawbacks of symbolic representations. Examples include classifying human intent in natural language and dialogues [33, 6], and in basic activities like walking or standing [27, 13, 20]; or predicting the intended directions of pedestrians and cars [5] in self-driving vehicles.

These problems deal with short sequences of observations. Here we are interested in inferring the long-term intention of a goal-oriented agent, for example, the end destination of a person traveling. From a cognitive standpoint, recognizing the intention behind such long-term goal-driven behaviors require the ability to reason somehow about a model of the world dynamics and project the observed agent into the future. That is, it requires to reverse the planning process [3].

End-to-end deep learning approaches to long-term intent recognition so far consist of casting the inference as a pattern recognition problem, using standard recurrent deep neural networks such as LSTM, without any component to project behaviors into the future [18, 1]. Yet, insight into cognitive models to intent recognition suggests that the ability to reason about the future is crucial to achieving effective recognition capability of long-term intents [3].

In this line of inquiry, we introduce a novel approach to intent recognition using a standard spatiotemporal deep neural network (STDNN) adjoined with a future projection module providing future insight in terms of gradients of costs that are concatenated to the input features of the network. The approach demonstrates improved performance compared to a baseline STDNN devoid of the future insight in a synthetic navigation domain. Interestingly, given that the future-insight features are handled as the other input features by the neural network, this approach is robust to imprecision, incompleteness, and errors that may be conveyed by the symbolic action model.

2.2 Intent Recognition as a Learning Problem

Following Sukthankar *et al.* [30], we formalize the intent recognition problem as a goal recognition problem. Given a sequence of observations $O = o_1, \ldots, o_n$ of an agent's behavior in his environment, and a set G of potential goals that he might pursue, the problem is to infer a posterior probability distribution across G, $P(G|O)$, representing the probabilities of the agent seeking these goals given the observations.

Intent recognition can be cast as a supervised learning problem as follows. Assuming that the agent's behavior is goal-oriented and that he pursues a single goal at a time, there must exist an optimal mapping f between O and G. Let us suppose we have access to a dataset $\mathcal{O}$ of sequences of observations of an agent and the corresponding true goal $g \in G$ he pursued. We can approximate f with a function f' parametrized by θ minimizing the number of erred predictions in the dataset. In other words, f' minimizes

$$L = \sum_{n=0}^{N} l(f'(O^n; \theta), g_{O^n})$$

where O^n is the n^{th} example, g_{O^n} its corresponding goal, N the size of our dataset and l is a loss function that is 0 when f' predicts accurately, and strictly positive otherwise.

We minimize the objective through stochastic gradient descent of the parameters of a deep neural network (DNN). In their simplest form, DNNs are chains of linear transformations of the input data [8]. Combining multiple of these transformations is what allows DNNs to perform end-to-end learning, i.e. learning an optimal representation of the raw data for the task at hand as well as the task itself. All the same, there exist variations specially designed to do so for some type of input. For instance, convolutional neural networks (CNNs) convolve locally-connected parameters organized in filters on spatial-wise organized data, such as images or video frames [14]. Recurrent neural networks (RNNs) process temporal data sequence-wise through an internal state memorizing the first elements. Long Short-Term Memory (LSTM) networks are a variation using multiple learned gates to cope with vanishing and exploding gradients affecting vanilla RNNs on longer sequences [10].

Since O is undeniably temporal, we benefit from the recurrence of the parameters in our architecture through LSTM cells. We also exploit the spatiality of grid worlds through convolutional operations, endowing the solution with insight on the connectivity of the grid positions. This results in a neural network capable of parsing spatiotemporal observation data, i.e an STDNN. To read the full details on the architectural choices, see the applicable section.

2.3 Future Projection for Intent Recognition

The particularity about the long-term goal-driven intent recognition problem is that the goal to predict is not observed until the end of the sequence. A useful property of a solution is then to be online – i.e. making multiple inferences incrementally during the observation process – and to be able to predict the goal as earlier as possible. Therefore, we cannot wait for the end of the observation process to make an inference about the pursued goal. In that case, it becomes helpful to be able to imagine the next steps that the agent will reasonably take and where it will take him, so as to make an enlightened prediction. That is, it requires the ability to project him into the future.

Future projection essentially requires two components: a model of the environment and a way to explore it. Since we rely on symbolic approaches for doing so, both these components are based on knowledge, which we provide as a domain theory and a planner. In the case of intent recognition problems though, knowledge about how the agent reasons is also necessary and will influence the choice of the planner that mimics the imagined agent's behavior.

Many first symbolic approaches did so by providing a plan library, which is an exhaustive engineered enumeration of all the possible plans of the agent [30]. We chose instead to borrow ideas from cost-based plan recognition algorithms that overcome the need for this library. The intuition is, assuming that the observed agent is rational – a.k.a cost-sensitive – he will be more likely to pursue the least-cost plan. To make a goal inference, an observer need only compare the observed plan of the agent with the optimal plan for some goal. If the two costs match, then this goal is plausible [22]. Therefore, only an optimal planner and a domain theory become necessary to make a goal inference.

Making this approach probabilistic accounts partially for potential divergences of the agent from the optimal behavior. For instance, Ramírez and Geffner [23] compute the goal inference using a Boltzmann distribution:

$$P(g|O_{0:t}) = \alpha \frac{1}{1 + \exp(\beta\Delta(s_0, g, O_{0:t}))} \tag{2.1}$$

where α is a normalisation factor, β is a temperature hyperparameter tuned according to the agent's assessed optimality, s_0 is the initial position, $g \in G$ is the goal we make inference on, and Δ is the following cost difference formula:

$$\Delta(s_0, g, O_{0:t}) = c(s_0, g, O_{0:t}) - c(s_0, g, \bar{O}_{0:t}) \tag{2.2}$$

where $c(s_0, g, O_{0:t})$ is the cost of the cheapest plan from s_0 to g complying to the observed actions in $O_{0:t}$, and $c(s_0, g, \bar{O}_{0:t})$ is the cheapest plan reaching g where at least one of the observed actions have not occurred.

Vered *et al.* [32] rather use a cost ratio to make a probabilistic inference:

$$P(g|O_{0:t}) = \alpha \frac{c(s_0, g)}{c(s_0, g, O_{0:t})} \tag{2.3}$$

where $c(s_0, g)$ is simply the cost of the optimal plan between s_o and g.

Masters and Sardiña [16] use a simpler cost difference formula accounting only for the initial and last observations:

$$\Delta(s_0, s_t, g) = c(s_t, g) - c(s_0, g) \tag{2.4}$$

where $c(s_t, g)$ is the cost of the optimal plan from the last observed position s_t to g. This method allows them to compute costs offline and store them into convenient costs maps for navigation domains, an idea that we borrowed for our application. They also suppose a Boltzmann probability distribution over this difference.

Even though these three methods are probabilistic, issues arise when the agent displays irrational behavior towards *all* goals [17]. Cost-based plan recognition algorithms postulate that the agent reflects at least more-or-less optimal behavior towards his true goal since it seeks the lowest-cost plan. In real-life situations though, this assumption may break for multiple reasons.

Let us take the example in figure 2.1, where an agent navigates in some environment. In that example, the agent is suboptimal towards all goals, since O is not on any optimal path to one of them [1]. Yet, this situation could realistically happen, if we imagine that the agent changed its mind about its goal, some paths are less desirable than others, or there are unseen obstacles. The point is, any violation of the knowledge that we have about the world and the agent (e.g deterministic, fully observable, uniform costs, etc.) can make him appear as irrational.

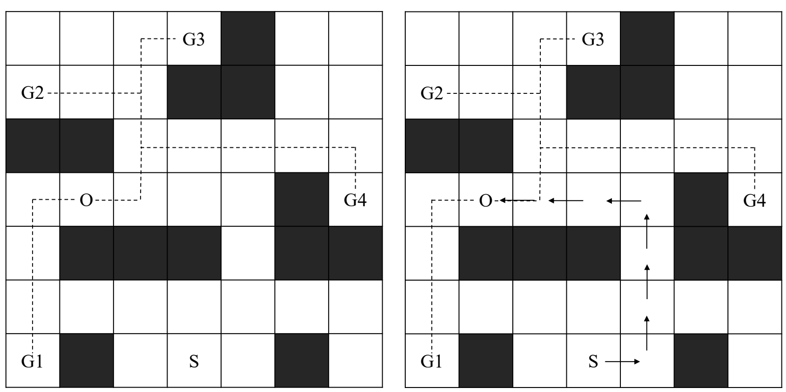

Figure 2.1 – Example of a suboptimal agent navigating a grid. S is his initial position, O is his last observed position, G1 to G4 are possible goals, and dashes are projected optimal paths. (Left) Without the observations in-between S and O, it is unclear what is the destination of the agent. (Right) With all observations (arrows), G1 now appears as a likely goal.

1. Following the definitions introduced by Masters and Sardina [17], the agent is strictly less rational, but not uniformly less rational.

Whether the agent is truly irrational or not, costs do not convey as useful information in these situations. Indeed, relying only on this knowledge to make goal inference makes the situation ambiguous and return counter-intuitive results. For instance, equation 2.4 ranks both G2 and G3 first (since $P(G|O)$ is maximal when Δ is minimal) followed by both G1 and G4. Since it relies only on two observations to make an inference (as depicted on the left pane of 2.1) crucial information residing in the other observations do not weigh in the decision. Indeed, looking at the right pane and knowing that the agent went right, it now seems reasonable to consider G1 as likely.

Yet interestingly enough, equations 2.2 and 2.3 make the same prediction, even though their cost formulas rely on all observations. This is because the information conveyed in the observations is reduced to 2 optimal costs for each goal, while the agent is *not* optimal.

We thus suggest relying also on the information disclosed in the observation data to make an inference. We hypothesize that the STDNN fed with both observations and plan costs can cope with possible inadequacies of the symbolic model by learning complementary features. The costs here hence only serve as an insight on the agent's possible future and are not the sole features on which goal inference depends.

In that sense, we designed a future projection module to generate the shortest cost-to-go of the agent from its current position to every position of the map [2] for every new observation. This results in 2D cost maps (pictured in figure 2.2) that are fed along with the observations to the STDNN. This gives it sufficient information about each speculated goal-achieving future situation and their evolution at each timestep.

2.4 Gradients of Costs

We gave our STDNN intuition about the agent's possible futures through optimal plan costs to perform insightful long-term intent recognition. Nonetheless, plan costs

2. Even though the costs-to-go of the goal positions only would have been sufficient, we felt that the network would benefit from the spatial information of the resulting cost map and generalize better to different goal configurations.

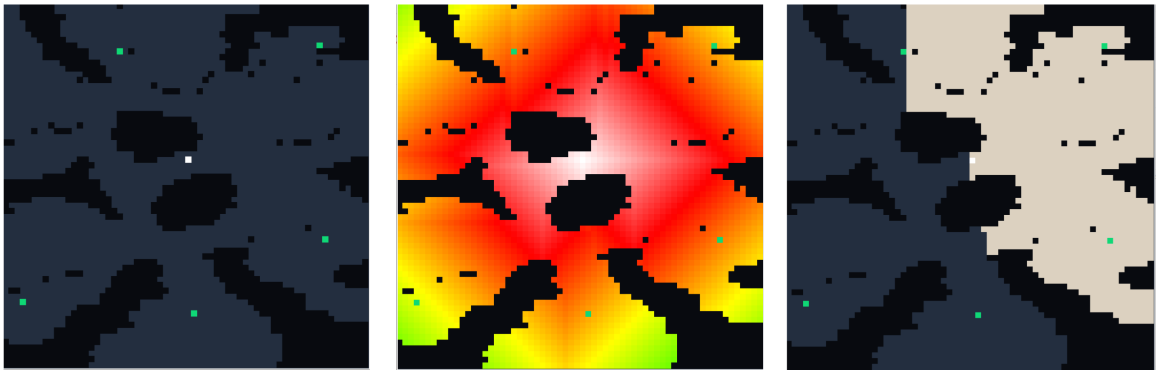

Figure 2.2 – Input features fed to the neural network. (Left) Observation of an agent that just moved right. The agent is the white pixel, and the potential goals are in green. The obstacles are black. (Middle) Corresponding costs map. Warmer colors indicate smaller costs. (Right) Corresponding differential costs map. The lighter color indicates positive values, while the other indicates negative.

are not a feature exploiting fully the knowledge we have about how to make smart goal inference.

Indeed, the important information provided from the costs is in their variation from timestep to timestep. If we could only see the last timestep and had no access to the history of the observations, the costs themselves would not give essential information. It might even bias the observer into thinking that lower costs are always better, which would result in a solution where the correct goal inference will be made only when the true goal becomes the closest.

A better feature is one that exploits the knowledge of the variation: that is, costs of the true goal get *lower* with time. We thus engineered a novel intent recognition feature by taking the partial derivative of the cost maps over time. The feature is computed as follows:

$$\frac{\partial c(s_t,(x,y))}{\partial t} = c(s_{t-1},(x,y)) - c(s_t,(x,y)) \tag{2.5}$$

where $c(s_t,(x,y))$ is the optimal cost from the agent's position s_t to (x,y) [3]. By taking

3. Since we consider discrete timesteps, we approximate the partial derivative from a single

the derivative for every (x, y), we obtain a differential cost map (shown in figure 2.2). Put simply, such a derivative gives a global idea about the current moving direction of the agent. By computing this map every timestep, we obtain the *gradient* of the costs generated for the sequence of observations O, denoted $grads(O)$.

The advantage of this new intent recognition feature over cost differences of the cost-based algorithms presented earlier is that it cuts less information about the observations and costs, and thus gives the STDNN more flexibility about the inference. This is crucial to keep the system robust against certain violations of the rationality assumption or other incorrectness about domain knowledge.

Taking again the example on figure 2.1, the STDNN can weigh each differential cost feature according to their neighboring values and their age in timesteps. Equations 2.2, 2.3 and 2.4 do not allow to do it in the time dimension, since they always compare to the initial projected future at s_0, while equation 2.5 at point t is function only of the two last timesteps. $grads(O)$ depends on the past computed differences as well, but the early ones do not affect the values of the latest ones. It is possible for the STDNN to *forget* past differential cost maps by giving them a smaller weight if it helps it to cope with the agent's past suboptimality by not taking into account early mistakes. All the same, the past differences can serve to avoid discarding a goal too early if the agent recently took a single suboptimal step for that goal. Even though they are produced by a potentially incorrect rationality assumption, gradients of costs comport all the necessary information for our intent inference solution to balance the past with the future.

2.5 Architectural Choices and Experimental Setup

We present herein the benchmark we used to evaluate the capacity of our approaches for intent recognition and detail our architectural choices.

timestep delta. We use a past point to avoid the formula to depend on future information.

2.5.1 Environments

The environments are a set of 28 maps taken from MovingAI[4]. All the maps were downscaled to 8x8, 16x16, 32x32 and 48x48 map sizes, to appreciate how the compared methods could scale up in more challenging problem sizes.

An intent recognition problem in this setting consists of predicting the goal destination of the agent in the grid. The agent's moves are limited to the four basic cardinal directions. Diagonal moves are not allowed, favoring the use of unit cost action[5]. The plan recognition problems were generated by repeatedly sampling 6 different non-obstacle positions on the maps, the first one being the start position and the others the possible destinations of the agent. The positions were sampled by making sure that goals were spaced enough to make the problems suitable for a comparing evaluation of the algorithms in online applications. Indeed, goals that are next to each other are impossible to distinguish by any method until the end of the observation sequence. The minimal space in-between goals for each problem size is in table 2.1.

Table 2.1 – Characteristics of the problems

	8x8	16x16	32x32	48x48
Minimal goal spacing	4	7	13	19
Mean path length	7.72	14.67	30.02	47.22
# Epochs	200	400	800	1500

The observations of the agent were generated using the A* algorithm to find a path between the start position and one randomly picked goal. Our A* algorithm was aided with what we define as an ϵ-over-estimating heuristic:

Definition 2.5.1 *An ϵ-over-estimating heuristic h is a heuristic that returns an admissible quantity h' in $1-\epsilon$ of cases, and $h'+\delta$ otherwise, where $\epsilon \in [0, 1]$ and $\delta > 0$.*

We used the Manhattan distance as the admissible heuristic h', an ϵ of 0.2 and a δ randomly sampled between 0 and 10. The agent is thus not guaranteed to be optimal

4. MovingAI Lab: https://movingai.com/

5. It is not a requirement of the solution, simply a design choice. Costs maps and $grads(O)$ can be computed over variable action costs as well.

and can be suboptimal up to 20% of the time. In practice, it is less than that, since the overestimation may still result in choosing the optimal path. Besides, the agent cannot trace back its steps on already visited nodes, since the rest of the process works as a standard search algorithm. This resulted in increasing path lengths with the problem size. The mean length of the generated paths is in table 2.1.

Each of the observations takes the form of an 8-channel bitmap bird view of the environment where each channel represents whether the grid cell is an obstacle, a walkable cell, the agent's observed position, or one of its possible goal destination, where each goal is attributed a different channel. This makes each goal distinctive from one another and favored output predictions using 5-dimension vectors.

2.5.2 Architecture

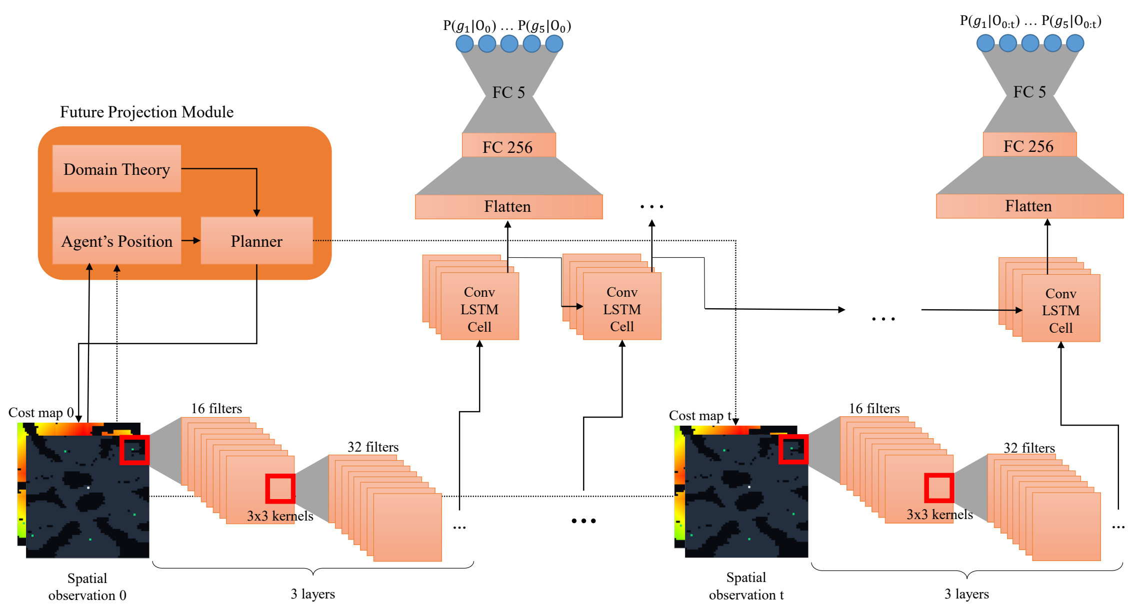

Figure 2.3 – Future projection module and spatiotemporal network architecture. Red squares represent convolution filters. FC is for Fully Connected. Optional max pooling layer is not displayed. The future projection module can return either costs maps or differential costs maps.

Our neural network architecture (figure 2.3) is composed of:

1. 3 spatial-wise convolutional layers consisting of 16, 32, and 64 3x3 filters respectively, with a stride of one, same padding, and each followed by a ReLU activation;
2. An optional 2x2 max-pooling layer for problems of size greater than 16x16 between the first and second layer;
3. A convolutional LSTM layer consisting of 32 3x3 filters per gate and 32 for the cell state (for a total 128 filters);
4. A densely connected layer of 256 units over the flattened output of the LSTM cell, followed by hyperbolic tangent (tanh) activation;
5. A final densely connected layer of 5 units followed by a softmax activation for goal inference.

Dropout [29] with a drop rate of 0.1 was applied in-between each parametrized layer. The network was trained using the categorical cross-entropy loss between the predictions and the supervised targets generated along the sequence of observations. Even though the goal never changes during a sequence, we output one goal prediction by observation and train on each produced output. This enables training with all possible partial sequences of a full observation sequence simultaneously, which comes useful for the online application of the solution. The network is trained for a certain number of epochs depending on the problem size (see table 2.1). Each epoch consists of 64 training iterations of mini-batches of size 32. The examples were generated in parallel to the training process. Since they were not saved, the network may never have seen the same example twice (even in-between epochs). On the other hand, the validation and test sets of 160 and 864 examples respectively were generated beforehand to ensure that all methods were tested against the same examples.

Finally, the network was optimized using the Adam optimization algorithm [12], with a learning rate of 1e-3, beta1 of 0.9 and beta2 of 0.999 respectively. Furthermore, the learning rate is gradually reduced with a factor of 0.9 every 10 epochs when a plateau in validation loss is detected, to a minimal value of 1e-5.

2.5.3 Costs Maps and Gradients Generation

Since the set of environments used was known and finite, it was possible for us to compute the cost maps offline and to store them before training and testing. The original 28 bitmaps from MovingAI – where nonzero values represent walls – were parsed to build the connection graphs. They were then passed to a succession of calls to the breadth-first search (BFS) algorithm for each of the positions of the map to generate the cost-to-go from that position. The resulting costs maps were stored in 28 4-dimension tensors, where the first 2 axes represent the start position, and the 2 last ones the end position. The process was repeated for all problem sizes.

As for gradients of costs, they were computed online by subtracting the current timestep cost map from the previous one, starting from the second observation. We passed down an all-zeros map as the derivative of the first timestep. In both cases, the resulting matrix was concatenated to the last observation, making it a 9-channel image input (see figure 2.3).

2.6 Experiments and Results

We tested the performance of our STDNN, augmented with costs maps or $grads(O)$, against the following baselines:

- STDNN on observation data only
- Masters and Sardiña [16]'s method (denoted MS)
- Ramírez and Geffner [23]'s method (denoted RG)
- Vered *et al.* [32]'s method (denoted VK)

Each of the STDNNs was trained on 12GB Pascal architecture NVIDIA GPUs. A single GPU per network instance was used for the first 2 problem sizes, and 2 GPUs for the larger sizes.

We measured the accuracy of each method at 10 different relative points in time. A measure at 0.1 indicates that only the first 10% of the observations were shown, while the last measure (at 1) was made on the full sequence up to the goal-achieving timestep. It is important to note that the accuracy metric we used only checks if the *first* goal with maximal probability matches the ground truth: in case of equality between k goals, there is only a theoretical $1/k$ chance that the matched goal will be the right one. This is to ease the comparison with the symbolic baselines since they often rank the goals with equal probabilities early on. We feel that this metric summarizes the findings in a visually concise and pleasant way.

We also compute the mean rationality measure of the agent proposed by Masters and Sardina [17] to see how much the agent diverges from optimal behavior. The metric is given by:

$$RM(G, O_{0:t}) = \max_{g \in G} \frac{c(s_0, g)}{c(s_0, g, O_{0:t})}$$

Results are shown on figure 2.4. For all graphs, we merged the curves of MS and RG since they had identical performances. On the easiest problem size, all STDNNs performed well, easily surpassing symbolic methods. This can be explained by the fact that at this size, the receptive field of the stacked convolutional layers easily covers the entire map, and the problems lengths are short enough for future projection not to be required. All STDNNs thus rapidly fit the agent's behavior, and even though it stayed relatively optimal, the symbolic methods could not keep with their performance.

The 16x16 problem size is already too challenging for the simple STDNN. It would probably have been possible to achieve similar performances with a deeper network; however, both augmented STDNNs were able to learn quite easily without the help of more layers. This suggests the features indeed gave insightful information about the problem as hypothesized. Again, they surpassed easily the performance of the symbolic approaches, while the mean rationality of the agent stayed over 95%.

On the last two problem sizes though, only the STDNN augmented with $grads(O)$ was able to keep with the performance of the symbolic algorithms, slightly outperforming MS, RG, and VK, but not significantly. This may be explained by the fact that the internal variance of the values in the cost maps was high, greatly impacting learning. As for the differential cost maps, the range of their values stay in a fixed interval no matter the problem size. This makes it the fitter feature to provide an intuition of the evolution of the possible goal-achieving futures of the intent recognition problem since the learning method can afford to scale to large problem sizes.

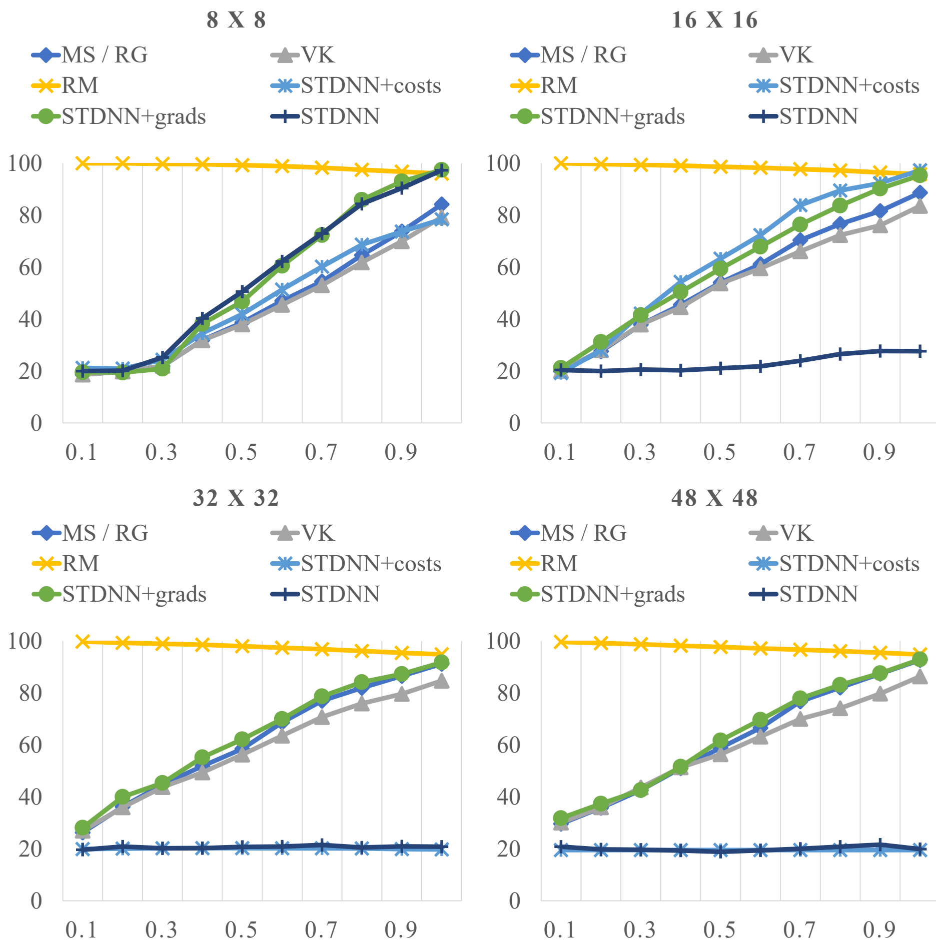

Figure 2.4 – Results for the accuracy (in percentage) of each method over the relative observation sequence length. RM is the mean rationality measure (in percentage).

2.6.1 Test with a Suboptimal Agent

In light of the previous results, we tested experimentally the example on figure 2.1 to validate our theoretical expectations. We used the STDNN augmented with $grads(O)$ trained on 8x8 maps, and adapted the 7x7 example by bordering it with obstacles and adding a fifth goal below S so as to not influence goal inference (since the first step of the agent is to go up, we were confident that no method would ever rank this goal as likely). Besides, we normalized the probabilities returned for the four goals of the example only.

We completed the observation sequence with a path towards G1. Results are reported in table 2.2. We set $\beta = 1$ in equation 2.1 [6]. As both Ramírez and Geffner [23] and Masters and Sardiña [16] reported the exact same probabilities, we merged them in the table.

The observation displayed on figure 2.1 is the 7th one. At this point, no method makes the correct inference. However, MS, RG and VK's probabilities stayed relatively the same as in the 6th step, while there was a significant drop in the probabilities of G2 and G3 returned by the network. Besides, STDNN now considers G1 as more likely than G4, at the opposite of the other methods. One step later, STDNN now makes the correct inference with mild confidence. MS and RG continue to presume it is either G2 or G3. Two steps away from G1, MS / RG now consider G1 as likely, but only as equally probable as G2 and G3, while STDNN's probability for G1 is now over 90%. VK never made the right inference, even one step away from the goal.

Table 2.2 – Probability results on the example of figure 2.1

	STDNN+grads (%)				**MS / RG (%)**				**VK (%)**				**RM**
	G1	G2	G3	G4	G1	G2	G3	G4	G1	G2	G3	G4	(%)
0	25.75	24.47	23.98	**25.81**	25.00	25.00	25.00	25.00	25.00	25.00	25.00	25.00	100.00
1	3.53	26.01	24.62	**45.85**	4.32	**31.89**	**31.89**	**31.89**	19.23	**26.92**	**26.92**	**26.92**	100.00
2	4.63	26.58	22.47	**46.32**	4.32	**31.89**	**31.89**	**31.89**	19.23	**26.92**	**26.92**	**26.92**	100.00
3	0.79	26.69	26.46	**46.06**	0.61	**33.13**	**33.13**	**33.13**	15.63	**28.13**	**28.13**	**28.13**	100.00
4	0.46	24.51	26.50	**48.52**	0.08	**33.31**	**33.31**	**33.31**	13.16	**28.95**	**28.95**	**28.95**	100.00
5	0.84	**47.13**	45.35	6.68	0.12	**46.78**	**46.78**	6.33	13.97	**30.73**	**30.73**	24.58	100.00
6	0.91	**57.64**	40.54	0.92	0.12	**49.49**	**49.49**	0.91	14.56	**32.04**	**32.04**	21.36	100.00
*7	18.82	**47.90**	30.26	3.02	0.90	**49.10**	**49.10**	0.90	16.88	**30.95**	**30.95**	21.22	83.33
8	**69.36**	14.23	13.44	2.97	6.28	**46.43**	**46.43**	0.85	19.07	**29.97**	**29.97**	20.98	71.43
9	**94.78**	1.34	2.86	1.02	**33.13**	**33.13**	**33.13**	0.61	21.15	**29.08**	**29.08**	20.68	62.50
10	**99.02**	0.15	0.60	0.23	**78.55**	10.63	10.63	0.19	23.12	**28.26**	**28.26**	20.35	55.56

We wish to point out that the network was never trained on this example, nor even this map configuration. Moreover, the rationality reported is way below what was seen in the previous experiments. This confirms the adaptability of our method for suboptimal behavior.

6. We in fact used the alternate non-sigmoidal probability distribution proposed by Masters and Sardina [17].

2.7 Related Work

The future projection capability is seeing a growing research interest from the deep learning community. The Predictron [26] learns to model rewards of a reinforcement learning environment to more accurately estimate the cumulative value of policies explored by a deep-learned approach. Imagination-augmented agents [21] that inspired our work, goes further by using model-based deep reinforcement learning ideas to imagine future projected trajectories to guide the exploration of a model-free deep policy learner in Sokoban, PacMan and other related games. Dosovitskiy and Koltun [7] transforms the standard reinforcement learning setting into a self-supervised one by attempting to predict action effects on measurements (such as altitude, health, etc.) and give desirable values of measurements to achieve. Ha and Schmidhuber [9] use variational auto-encoders to simulate world models of Doom and other games and an evolutionary algorithm to learn from these simulations. Ke *et al.* [11] effectively learn to predict long-term future using improved LSTM architectures and show how it helps in various planning tasks either deep learned by imitation or from reinforcement. While these approaches have been tried on multiple problems involving long-term reasoning, long-term intent recognition is not one of them. Another aspect is that they all chose to learn future projection, while we rely on symbolic models and planners. This enabled us to achieve impressive results on challenging problem sizes using a simpler architecture and fewer data.

For intent recognition problems, others have tried to combine symbolic ideas with deep learning. Asai and Fukunaga [2] used autoencoders to transform the observations into a latent space, from which they extract a model they feed to a planner that compute costs for intent inference. Pereira *et al.* [19] use DNNs to learn nominal models over continuous domains in an attempt to resolve bias introduced when engineering them, and also make symbolic goal inference with the help of plan costs. Bisson *et al.* [4] used provided plan libraries in the form of a context-free grammar to design recursive neural networks that learn the probabilistic model for plan recognition. The distinction between these works and ours is that either the domain knowledge or the probabilistic model is learned, but not both.

There also exist approaches that use a complete DL pipeline. Min *et al.* [18] used multimodal LSTM networks to predict the next goal of the player in the CRYSTAL ISLAND game. Amado *et al.* [1] extends the work of Asai and Fukunaga [2] by replacing all symbolic parts with an LSTM network. However, neither entail a way to perform future projection and were trained and tested on problems from a single environment, while we generalize to multiple grid environments.

Solutions to cope with erroneous knowledge have also been researched in the last few years. Sohrabi *et al.* [28] compute the top-k plans for each goal and use their degree of compliance with the observations as well as their cost to deal with noisy and missing observations. This makes it potentially more robust to suboptimal behaviors, but with a significantly higher computation cost. Masters and Sardina [17] designed a way to measure the rationality of the agent and vary the β parameter of equation 2.1 accordingly. In the case of irrational behavior, the probabilities are flattened out, but their order and the final prediction do not change. Finally, Pereira *et al.* [19] deal with erroneous domain theory, but still assumes the agent's rationality and use optimal plan costs to make an inference. We are thus confident that our solution is a step further towards real-life application since it can learn to make accurate predictions from incorrect domain knowledge, such as from suboptimal behaviors.

2.8 Conclusion

We presented an innovative solution to intent recognition by combining deep learning and symbolic AI strengths: the ability to adapt from experiences with the generalizable capability to project the observed agent into the future, which is inherent to long-term goal-driven intent recognition. Our solution augmented with our novel gradients of costs feature easily outperforms the non-augmented deep learned solution and even surpasses state-of-the-art pure symbolic AI methods. We demonstrated that our approach can predict the goal of the agent faster than pure cost-based algorithms in situations where our assumptions about the agent's behavior are wrong (i.e. when he is suboptimal), hence proving its robustness.

Thanks to that property, and the fact that cost maps computation and neural network training can be performed offline, we are hopeful to be able to adapt this solution to online real-world situations. It could be relevant to continue to research its capability by extending it to other challenging intent recognition domains, exploring in the meantime more general DNN architectures such as graph neural networks [24].

2.9 Acknowledgements

This work was supported by grants from the Natural Sciences and Engineering Research Council (NSERC) of Canada and the *Fonds de recherche du Québec – Nature et technologies* (FRQNT), and also by Compute Canada for the computing resources they provided. We are also thankful to Hugo Larochelle, whose feedback and advice inspired major ideas of this paper.

References

[1] L. Amado, J. P. Aires, R. F. Pereira, M. C. Magnaguagno, R. Granada, F. Meneguzzi, "LSTM-Based Goal Recognition in Latent Space," *CoRR*, vol. abs/1808.05249, 2018. [Online]. Available: http://arxiv.org/abs/1808.05249

[2] M. Asai A. Fukunaga, "Classical Planning in Deep Latent Space: Bridging the Subsymbolic-Symbolic Boundary," in *AAAI 2018*, 2018. [Online]. Available: https://aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16302

[3] C. L. Baker, R. Saxe, J. B. Tenenbaum, "Action understanding as inverse planning," *Cognition 2009*, vol. 113, no. 3, pp. 329–349, 2009.

[4] F. Bisson, H. Larochelle, F. Kabanza, "Using a Recursive Neural Network to Learn an Agent's Decision Model for Plan Recognition," in *IJCAI 2015*, 2015, pp. 918–924.

[5] S. Casas, W. Luo, R. Urtasun, "IntentNet: Learning to Predict Intention from Raw Sensor Data," in *Proceedings of The 2nd Conference on Robot Learning*, vol. 87. PMLR, 29–31 Oct 2018, pp. 947–956. [Online]. Available: http://proceedings.mlr.press/v87/casas18a.html

REFERENCES

[6] C. Chen, X. Zhang, S. Ju, C. Fu, C. Tang, J. Zhou, X. Li, "AntProphet: an Intention Mining System behind Alipay's Intelligent Customer Service Bot," in *IJCAI 2019*, 08 2019, pp. 6497–6499.

[7] A. Dosovitskiy V. Koltun, "Learning to Act by Predicting the Future," in *ICLR 2017*, 2017. [Online]. Available: https://openreview.net/forum?id=rJLS7qKel

[8] I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning*. MIT Press, 2016.

[9] D. Ha J. Schmidhuber, "Recurrent World Models Facilitate Policy Evolution," in *NeurIPS 2018*, 2018, pp. 2450–2462. [Online]. Available: http://papers.nips.cc/paper/7512-recurrent-world-models-facilitate-policy-evolution.pdf

[10] S. Hochreiter J. Schmidhuber, "Long Short-term Memory," *Neural computation*, vol. 9, pp. 1735–80, 12 1997.

[11] N. R. Ke, A. Singh, A. Touati, A. Goyal, Y. Bengio, D. Parikh, D. Batra, "Modeling the Long Term Future in Model-Based Reinforcement Learning," in *ICLR 2019*, 2019. [Online]. Available: https://openreview.net/forum?id=SkgQBn0cF7

[12] D. P. Kingma J. Ba, "Adam: A Method for Stochastic Optimization," *CoRR*, vol. abs/1412.6980, 2014.

[13] Y. Kong Y. Fu, "Human Action Recognition and Prediction: A Survey," *CoRR*, vol. abs/1806.11230, 2018. [Online]. Available: http://arxiv.org/abs/1806.11230

[14] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, 1998, pp. 2278–2324.

[15] P. Masters S. Sardiña, "Cost-Based Goal Recognition in Navigational Domains," *JAIR*, vol. 64, pp. 197–242, 2019. [Online]. Available: https://doi.org/10.1613/jair.1.11343

[16] P. Masters S. Sardiña, "Cost-Based Goal Recognition in Navigational Domains," *JAIR*, vol. 64, pp. 197–242, 2019. [Online]. Available: https://doi.org/10.1613/jair.1.11343

[17] P. Masters S. Sardina, "Goal Recognition for Rational and Irrational Agents," in *AAMAS 2019*, Richland, SC, 2019, pp. 440–448. [Online]. Available: http://dl.acm.org/citation.cfm?id=3306127.3331725

[18] W. Min, B. Mott, J. Rowe, R. Taylor, E. Wiebe, K. Boyer, J. Lester, "Multimodal Goal Recognition in Open-World Digital Games," in *AAAI 2017*, 2017. [Online]. Available: https://aaai.org/ocs/index.php/AIIDE/AIIDE17/paper/view/15910

[19] R. F. Pereira, M. Vered, F. Meneguzzi, M. Ramírez, "Online Probabilistic Goal Recognition over Nominal Models," in *IJCAI 2019*, 2019, pp. 5547–5553. [Online]. Available: https://doi.org/10.24963/ijcai.2019/770

[20] H. Qian, S. J. Pan, B. Da, C. Miao, "A Novel Distribution-Embedded Neural Network for Sensor-Based Activity Recognition," in *IJCAI 2019*, 2019, pp. 5614–5620. [Online]. Available: https://doi.org/10.24963/ijcai.2019/779

[21] S. Racanière, T. Weber, D. P. Reichert, L. Buesing, A. Guez, D. J. Rezende, A. P. Badia, O. Vinyals, N. Heess, Y. Li, R. Pascanu, P. W. Battaglia, D. Hassabis, D. Silver, D. Wierstra, "Imagination-Augmented Agents for Deep Reinforcement Learning," in *NIPS 2017*, 2017, pp. 5690–5701. [Online]. Available: http://papers.nips.cc/paper/7152-imagination-augmented-agents-for-deep-reinforcement-learning

[22] M. Ramírez H. Geffner, "Plan Recognition as Planning," in *IJCAI 2009*, 2009, pp. 1778–1783.

[23] M. Ramírez H. Geffner, "Probabilistic Plan Recognition Using Off-the-Shelf Classical Planners," in *AAAI 2010*, 2010.

[24] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, G. Monfardini, "The Graph Neural Network Model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, Jan 2009.

[25] C. F. Schmidt, N. S. Sridharan, J. L. Goodson, "The Plan Recognition Problem: An Intersection of Psychology and Artificial Intelligence," *Artificial Intelligence*, vol. 11, no. 1-2, pp. 45–83, 1978.

REFERENCES

[26] D. Silver, H. van Hasselt, M. Hessel, T. Schaul, A. Guez, T. Harley, G. Dulac-Arnold, D. P. Reichert, N. C. Rabinowitz, A. Barreto, T. Degris, "The Predictron: End-To-End Learning and Planning," in *ICML 2017*, 2017, pp. 3191–3199. [Online]. Available: http://proceedings.mlr.press/v70/silver17a.html

[27] K. Simonyan A. Zisserman, "Two-Stream Convolutional Networks for Action Recognition in Videos," in *NIPS 2014* Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, K. Q. Weinberger, editors, 2014, pp. 568–576. [Online]. Available: http://papers.nips.cc/paper/5353-two-stream-convolutional-networks-for-action-recognition-in-videos.pdf

[28] S. Sohrabi, A. V. Riabov, O. Udrea, "Plan Recognition As Planning Revisited," in *IJCAI 2016*, ser. IJCAI'16. AAAI Press, 2016, pp. 3258–3264. [Online]. Available: http://dl.acm.org/citation.cfm?id=3061053.3061077

[29] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[30] G. Sukthankar, C. Geib, H. H. Bui, D. Pynadath, R. P. Goldman, *Plan, Activity, and Intent Recognition: Theory and Practice*, 1st edition. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2014.

[31] M. Tomasello, M. Carpenter, J. Call, T. Behne, H. Moll, "Understanding and sharing intentions: The origins of cultural cognition," *Behavioral and Brain Sciences*, vol. 28, no. 5, p. 675–691, 2005.

[32] M. Vered, G. Kaminka, S. Biham, "Online goal recognition through mirroring: humans and agents," in *Fourth Annual Conference on Advances in Cognitive Systems*, 2016.

[33] T. Wen, Y. Miao, P. Blunsom, S. J. Young, "Latent Intention Dialogue Models," in *ICML 2017*, 2017, pp. 3732–3741. [Online]. Available: http://proceedings.mlr.press/v70/wen17a.html

Conclusion

Ce mémoire a exploré l'application de l'apprentissage profond au problème de reconnaissance de plan, un problème fondamental en intelligence artificielle qui reste toujours irrésolu. Or, cette exploration nous a permis de progresser vers sa résolution.

Dans le premier chapitre, nous avons comparé la performance de plusieurs architectures neuronales, comme les réseaux de neurones à convolutions, les réseaux récurrents à longue mémoire à court-terme et les réseaux complètement connectés, à celle d'algorithmes basés sur les coûts comme les algorithmes de Ramírez et Geffner [39] et de Masters et Sardiña [27]. Les réseaux de neurones ont surpassé les approches symboliques dans presque tous les domaines référentiels étudiés, mais ne pouvaient pas généraliser à plusieurs environnements d'un même domaine – autrement dit, ils nécessitaient d'être entraînés à nouveau pour un nouvel environnement et même une nouvelle configuration de buts dans cet environnement.

Dans le deuxième chapitre, nous avons alimenté un réseau de neurones spatiotemporel avec des caractéristiques issues de connaissances symboliques, lui offrant ainsi des notions cruciales sur le problème de reconnaissance d'intention à résoudre. La caractéristique, dénommée *gradients de coûts*, a permis au réseau de surpasser toutes les techniques de base auxquelles il s'est comparé, qu'elles soient purement apprises ou symboliques. De plus, le réseau entraîné était en mesure de généraliser sa performance à plusieurs cartes, tout en pouvant s'adapter au comportement sous-optimal de l'agent.

Pour l'instant, l'apprentissage profond a seulement été étudié dans des cas où l'agent observé est seul à agir dans un environnement déterministe complètement observable. Il serait intéressant d'évaluer son potentiel dans des situations où certaines de ces contraintes sont relâchées, par exemple dans un contexte où l'observateur et l'agent observé interagissent dans un même environnement, soit de manière coopérative ou conflictuelle. Ce type d'univers s'inscrit dans le domaine des jeux à deux joueurs. Une technique étudiée en théorie des jeux qui s'apparente justement à la reconnaissance d'intention est la modélisation de l'opposant [19]. Il serait intéressant d'analyser comment l'apprentissage profond peut exploiter cette technique pour reconnaître l'intention de son adversaire et le contrer, et comment un adversaire qui apprendrait lui aussi adapterait sa stratégie en conséquence, par exemple grâce à la tromperie.

Il serait d'autant plus pertinent d'appliquer l'apprentissage profond à des problèmes de reconnaissance d'intention issus de la vie réelle. Un des motifs premiers à l'étudier était justement de pouvoir éventuellement l'utiliser à cette fin, puisqu'il a démontré sa capabilité à traiter de grandes quantités de données brutes à haute dimensionnalité telles que des images et vidéos. Grâce aux gradients de coûts, les réseaux de neurones profonds sont d'autant plus près d'y arriver, puisque les gradients leur permettent de généraliser à plusieurs environnements issus d'un même domaine. Une autre piste pour la généralisation qui pourrait complémenter les gradients de coûts est l'apprentissage par transfert [18] et le méta-apprentissage [40]. Le premier permet de transférer l'apprentissage à un environnement non vu en fixant les poids des premières couches et en entraînant à nouveau les dernières couches sur peu d'exemples du nouvel environnement. Le deuxième, quant à lui, permet d'apprendre une initialisation des poids qui minimise le nombre d'exemples requis à l'apprentissage d'un nouvel environnement.

L'inconvénient des gradients de coûts est qu'il nécessite toujours de concevoir un modèle de l'environnement pour utiliser les planificateurs symboliques. De plus, l'approche ne s'applique qu'à la navigation dans une grille en deux dimensions pour le moment. Il serait profitable dans un premier temps de pouvoir généraliser l'approche à des domaines arbitraires en explorant les réseaux de neurones graphiques [42].

Ensuite, l'on pourrait imaginer que les connaissances qui ont servi à générer les gradients de coûts sont intégrées directement à une nouvelle architecture de réseau, afin qu'il sache lui-même les générer ou générer une caractéristique proche et représentative du problème. C'est en fait ce qui passe avec les réseaux de neurones à convolutions dont les premières couches génèrent des caractéristiques proches des gradients d'images, conçus pour les premières applications de vision par ordinateur. *Value Iteration Networks* [50] est un exemple d'architecture intégrant l'algorithme *Value Iteration*, lui permettant d'apprendre à estimer la valeur accumulée de manière similaire à l'algorithme et ainsi orienter le choix de sa politique. Les *Predictron* [45] sont quant à eux conçus pour modéliser une fonction de récompense qui facilite par après l'apprentissage de la valeur accumulée. De manière générale, plusieurs recherches visent à apprendre à mieux modéliser le futur [37, 10, 22] et apprendre de meilleurs plans ou stratégies issus de ces simulations imaginées [37, 10, 17]. Une autre avenue est d'utiliser l'algorithme symbolique en tant que superviseur pour l'entraînement des réseaux. Cette idée a été exploitée par AlphaZero [44] et Groshev *et al.* [16] qui ont utilisé des trajectoires générées par des algorithmes de recherche pour entraîner des réseaux à trouver de meilleurs plans et mieux estimer leur valeur. Le réseau est ensuite utilisé en tant qu'heuristique pour guider la recherche. Bien que ces idées aient seulement été appliquées à la planification, la résolution de jeux ou l'apprentissage par renforcement pour l'instant, elles pourraient aussi bien être étendues au problème de reconnaissance d'intention.

Bibliographie

[1] L. Amado, J. P. Aires, R. F. Pereira, M. C. Magnaguagno, R. Granada, F. Meneguzzi, « LSTM-Based Goal Recognition in Latent Space, » *CoRR*, vol. abs/1808.05249, 2018. Disponible à http://arxiv.org/abs/1808.05249

[2] M. Asai A. Fukunaga, « Classical Planning in Deep Latent Space : Bridging the Subsymbolic-Symbolic Boundary, » dans *AAAI 2018*, 2018. Disponible à https://aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16302

[3] C. L. Baker, R. Saxe, J. B. Tenenbaum, « Action understanding as inverse planning, » *Cognition 2009*, vol. 113, no. 3, pp. 329–349, 2009.

[4] F. Bisson, H. Larochelle, F. Kabanza, « Using a Recursive Neural Network to Learn an Agent's Decision Model for Plan Recognition, » dans *IJCAI 2015*, 2015, pp. 918–924.

[5] H. H. Bui, S. Venkatesh, G. A. W. West, « Policy Recognition in the Abstract Hidden Markov Model, » *JAIR*, vol. 17, pp. 451–499, 2002.

[6] S. Casas, W. Luo, R. Urtasun, « IntentNet : Learning to Predict Intention from Raw Sensor Data, » dans *Proceedings of The 2nd Conference on Robot Learning*, vol. 87. PMLR, 29–31 Oct 2018, pp. 947–956. Disponible à http://proceedings.mlr.press/v87/casas18a.html

[7] E. Charniak R. P. Goldman, « A Bayesian Model of Plan Recognition, » *Artif. Intell.*, vol. 64, no. 1, pp. 53–79, 1993. Disponible à https://doi.org/10.1016/0004-3702(93)90060-O

[8] C. Chen, X. Zhang, S. Ju, C. Fu, C. Tang, J. Zhou, X. Li, « AntProphet : an Intention Mining System behind Alipay's Intelligent Customer Service Bot, » dans *IJCAI 2019*, 08 2019, pp. 6497–6499.

[9] G. F. Cooper, « The Computational Complexity of Probabilistic Inference Using Bayesian Belief Networks (Research Note), » *Artificial Intelligence*, vol. 42, no. 2-3, pp. 393–405, mars 1990. Disponible à http://dx.doi.org/10.1016/0004-3702(90)90060-D

[10] A. Dosovitskiy V. Koltun, « Learning to Act by Predicting the Future, » dans *ICLR 2017*, 2017. Disponible à https://openreview.net/forum?id=rJLS7qKel

[11] Y. E.-Martín, M. D. R.-Moreno, D. E. Smith, « A Fast Goal Recognition Technique Based on Interaction Estimates, » dans *IJCAI 2015*, 2015, pp. 761–768.

[12] C. W. Geib R. P. Goldman, « Requirements for Plan Recognition in Network Security Systems, » dans *International Symposium on Recent Advances in Intrusion Detection 2002*, 2002.

[13] C. W. Geib R. P. Goldman, « A probabilistic plan recognition algorithm based on plan tree grammars, » *Artificial Intelligence*, vol. 173, no. 11, pp. 1101–1132, 2009.

[14] I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning.* MIT Press, 2016.

[15] R. Granada, R. Pereira, J. Monteiro, R. Barros, D. Ruiz, F. Meneguzzi, « Hybrid Activity and Plan Recognition for Video Streams, » dans *AAAI 2017*, 2017.

[16] E. Groshev, M. Goldstein, A. Tamar, S. Srivastava, P. Abbeel, « Learning Generalized Reactive Policies Using Deep Neural Networks, » dans *ICAPS 2018*, 2018, pp. 408–416.

[17] D. Ha J. Schmidhuber, « Recurrent World Models Facilitate Policy Evolution, » dans *NeurIPS 2018*, 2018, pp. 2450–2462. Disponible à http://papers.nips.cc/paper/7512-recurrent-world-models-facilitate-policy-evolution.pdf

[18] D. Hana, Q. Liu, W. Fan, « A New Image Classification Method Using CNN transfer learning and Web Data Augmentation, » *Expert Systems with Applications*, vol. 95, 11 2017.

[19] H. He, J. Boyd-Graber, K. Kwok, H. Daumé, III, « Opponent Modeling in Deep Reinforcement Learning, » dans *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, série ICML'16. JMLR.org, 2016, pp. 1804–1813. Disponible à http://dl.acm.org/citation.cfm?id=3045390.3045581

[20] S. Hochreiter J. Schmidhuber, « Long Short-term Memory, » *Neural computation*, vol. 9, pp. 1735–80, 12 1997.

[21] J. Hou, X. Wu, J. Chen, J. Luo, Y. Jia, « Unsupervised Deep Learning of Mid-Level Video Representation for Action Recognition, » dans *AAAI 2018*, 2018, pp. 6910–6917.

[22] N. R. Ke, A. Singh, A. Touati, A. Goyal, Y. Bengio, D. Parikh, D. Batra, « Modeling the Long Term Future in Model-Based Reinforcement Learning, » dans *ICLR 2019*, 2019. Disponible à https://openreview.net/forum?id=SkgQBn0cF7

[23] D. P. Kingma J. Ba, « Adam : A Method for Stochastic Optimization, » *CoRR*, vol. abs/1412.6980, 2014.

[24] Y. Kong Y. Fu, « Human Action Recognition and Prediction : A Survey, » *CoRR*, vol. abs/1806.11230, 2018. Disponible à http://arxiv.org/abs/1806.11230

[25] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, « Gradient-based learning applied to document recognition, » dans *Proceedings of the IEEE*, 1998, pp. 2278–2324.

[26] P. Masters S. Sardiña, « Cost-Based Goal Recognition for Path-Planning, » dans *AAMAS 2017*, 2017, pp. 750–758.

[27] P. Masters S. Sardiña, « Cost-Based Goal Recognition in Navigational Domains, » *JAIR*, vol. 64, pp. 197–242, 2019. Disponible à https://doi.org/10.1613/jair.1.11343

[28] P. Masters S. Sardiña, « Cost-Based Goal Recognition in Navigational Domains, » *JAIR*, vol. 64, pp. 197–242, 2019. Disponible à https://doi.org/10.1613/jair.1.11343

[29] P. Masters S. Sardina, « Goal Recognition for Rational and Irrational Agents, » dans *AAMAS 2019*, Richland, SC, 2019, pp. 440–448. Disponible à http://dl.acm.org/citation.cfm?id=3306127.3331725

[30] W. Min, E. Ha, J. P. Rowe, B. W. Mott, J. C. Lester, « Deep Learning-Based Goal Recognition in Open-Ended Digital Games, » dans *AIIDE 2014*, 2014.

[31] W. Min, B. W. Mott, J. P. Rowe, B. Liu, J. C. Lester, « Player Goal Recognition in Open-World Digital Games with Long Short-Term Memory Networks, » dans *IJCAI 2016*, 2016, pp. 2590–2596.

[32] W. Min, B. Mott, J. Rowe, R. Taylor, E. Wiebe, K. Boyer, J. Lester, « Multimodal Goal Recognition in Open-World Digital Games, » dans *AAAI 2017*, 2017. Disponible à https://aaai.org/ocs/index.php/AIIDE/AIIDE17/paper/view/15910

[33] E. Pacherie, « The phenomenology of action : A conceptual framework, » *Cognition*, vol. 107, no. 1, pp. 179–217, 2008. Disponible à https://jeannicod.ccsd.cnrs.fr/ijn_00352527

[34] R. F. Pereira, N. Oren, F. Meneguzzi, « Landmark-Based Heuristics for Goal Recognition, » dans *AAAI 2017*, 2017, pp. 3622–3628.

[35] R. F. Pereira, M. Vered, F. Meneguzzi, M. Ramírez, « Online Probabilistic Goal Recognition over Nominal Models, » dans *IJCAI 2019*, 2019, pp. 5547–5553. Disponible à https://doi.org/10.24963/ijcai.2019/770

[36] H. Qian, S. J. Pan, B. Da, C. Miao, « A Novel Distribution-Embedded Neural Network for Sensor-Based Activity Recognition, » dans *IJCAI 2019*, 2019, pp. 5614–5620. Disponible à https://doi.org/10.24963/ijcai.2019/779

[37] S. Racanière, T. Weber, D. P. Reichert, L. Buesing, A. Guez, D. J. Rezende, A. P. Badia, O. Vinyals, N. Heess, Y. Li, R. Pascanu,

P. W. Battaglia, D. Hassabis, D. Silver, D. Wierstra, « Imagination-Augmented Agents for Deep Reinforcement Learning, » dans *NIPS 2017*, 2017, pp. 5690–5701. Disponible à http://papers.nips.cc/paper/7152-imagination-augmented-agents-for-deep-reinforcement-learning

[38] M. Ramírez H. Geffner, « Plan Recognition as Planning, » dans *IJCAI 2009*, 2009, pp. 1778–1783.

[39] M. Ramírez H. Geffner, « Probabilistic Plan Recognition Using Off-the-Shelf Classical Planners, » dans *AAAI 2010*, 2010.

[40] S. Ravi H. Larochelle, « Optimization as a Model for Few-Shot Learning, » dans *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017. Disponible à https://openreview.net/forum?id=rJY0-Kcll

[41] A. Sadilek H. A. Kautz, « Recognizing Multi-Agent Activities from GPS Data, » dans *AAAI 2010*, 2010. Disponible à http://www.aaai.org/ocs/index.php/AAAI/AAAI10/paper/view/1603

[42] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, G. Monfardini, « The Graph Neural Network Model, » *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, Jan 2009.

[43] C. F. Schmidt, N. S. Sridharan, J. L. Goodson, « The Plan Recognition Problem : An Intersection of Psychology and Artificial Intelligence, » *Artificial Intelligence*, vol. 11, no. 1-2, pp. 45–83, 1978.

[44] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. P. Lillicrap, K. Simonyan, D. Hassabis, « Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm, » *CoRR*, vol. abs/1712.01815, 2017. Disponible à http://arxiv.org/abs/1712.01815

[45] D. Silver, H. van Hasselt, M. Hessel, T. Schaul, A. Guez, T. Harley, G. Dulac-Arnold, D. P. Reichert, N. C. Rabinowitz, A. Barreto, T. Degris, « The

Predictron : End-To-End Learning and Planning, » dans *ICML 2017*, 2017, pp. 3191–3199. Disponible à http://proceedings.mlr.press/v70/silver17a.html

[46] K. Simonyan A. Zisserman, « Two-Stream Convolutional Networks for Action Recognition in Videos, » dans *NIPS 2014* Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, K. Q. Weinberger, éditeurs, 2014, pp. 568–576. Disponible à http://papers.nips.cc/paper/5353-two-stream-convolutional-networks-for-action-recognition-in-videos.pdf

[47] S. Sohrabi, A. V. Riabov, O. Udrea, « Plan Recognition As Planning Revisited, » dans *IJCAI 2016*, série IJCAI'16. AAAI Press, 2016, pp. 3258–3264. Disponible à http://dl.acm.org/citation.cfm?id=3061053.3061077

[48] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, « Dropout : a simple way to prevent neural networks from overfitting, » *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[49] G. Sukthankar, C. Geib, H. H. Bui, D. Pynadath, R. P. Goldman, *Plan, Activity, and Intent Recognition : Theory and Practice*, 1st édition. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., 2014.

[50] A. Tamar, S. Levine, P. Abbeel, Y. Wu, G. Thomas, « Value Iteration Networks, » dans *NIPS 2016*, 2016, pp. 2146–2154.

[51] A. Tamar, Y. Wu, G. Thomas, S. Levine, P. Abbeel, « Value Iteration Networks, » dans *IJCAI 2017*, 2017, pp. 4949–4953.

[52] M. Tomasello, M. Carpenter, J. Call, T. Behne, H. Moll, « Understanding and sharing intentions : The origins of cultural cognition, » *Behavioral and Brain Sciences*, vol. 28, no. 5, p. 675–691, 2005.

[53] M. Vered G. A. Kaminka, « Heuristic Online Goal Recognition in Continuous Domains, » dans *IJCAI 2017*, 2017, pp. 4447–4454.

[54] M. Vered, G. Kaminka, S. Biham, « Online goal recognition through mirroring : humans and agents, » dans *Fourth Annual Conference on Advances in Cognitive Systems*, 2016.

[55] B. Volz, K. Behrendt, H. Mielenz, I. Gilitschenski, R. Siegwart, J. Nieto, « A data-driven approach for pedestrian intention estimation, » dans *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, Nov 2016, pp. 2607–2612.

[56] J. Wang, Y. Chen, S. Hao, X. Peng, H. Lisha, « Deep Learning for Sensor-based Activity Recognition : A Survey, » *Pattern Recognition Letters*, 07 2017.

[57] F. Warneken M. Tomasello, « Altruistic helping in human infants and young chimpanzees, » *Science*, vol. 311, pp. 1301–1303, 2006.

[58] F. Warneken, F. Chen, M. Tomasello, « Cooperative activities in young children and chimpanzees, » *Child Development*, vol. 77, no. 3, pp. 640–663, 2006.

[59] N. Watters, A. Tacchetti, T. Weber, R. Pascanu, P. Battaglia, D. Zoran, « Visual Interaction Networks, » *CoRR*, vol. abs/1706.01433, 2017.

[60] T. Wen, Y. Miao, P. Blunsom, S. J. Young, « Latent Intention Dialogue Models, » dans *ICML 2017*, 2017, pp. 3732–3741. Disponible à http://proceedings.mlr.press/v70/wen17a.html

[61] J. Wu, A. Osuntogun, T. Choudhury, M. Philipose, J. M. Rehg, « A Scalable Approach to Activity Recognition based on Object Use, » dans *ICCV 2007*, 2007.

[62] S. Yan, Y. Teng, J. S. Smith, B. Zhang, « Driver behavior recognition based on deep convolutional neural networks, » dans *ICNC-FSKD 2016*, 2016, pp. 636–641.