Teaching Agents how to Map: Spatial Reasoning for Multi-Object Navigation

Pierre Marza[1]
pierre.marza@insa-lyon.fr
Laetitia Matignon[2]
laetitia.matignon@univ-lyon1.fr
Olivier Simonin[3]
olivier.simonin@insa-lyon.fr
Christian Wolf[1]
christian.wolf@insa-lyon.fr

[1] LIRIS, UMR CNRS 5205
Université de Lyon, INSA Lyon
Villeurbanne, France

[2] LIRIS, UMR CNRS 5205
Université de Lyon, Univ. Lyon 1
Villeurbanne, France

[3] CITI Lab
Université de Lyon, INSA Lyon
INRIA Chroma team
Villeurbanne, France

Abstract

In the context of visual navigation, the capacity to map a novel environment is necessary for an agent to exploit its observation history in the considered place and efficiently reach known goals. This ability can be associated with spatial reasoning, where an agent is able to perceive spatial relationships and regularities, and discover object affordances. In classical Reinforcement Learning (RL) setups, this capacity is learned from reward alone. We introduce supplementary supervision in the form of auxiliary tasks designed to favor the emergence of spatial perception capabilities in agents trained for a goal-reaching downstream objective. We show that learning to estimate metrics quantifying the spatial relationships between an agent at a given location and a goal to reach has a high positive impact in Multi-Object Navigation settings. Our method significantly improves the performance of different baseline agents, that either build an explicit or implicit representation of the environment, even matching the performance of incomparable oracle agents taking ground-truth maps as input.

1 Introduction

Navigating in a previously unseen environment requires different abilities, among which is mapping, i.e. the capacity to build a representation of the environment and its affordances. The agent can then reason on this map and act efficiently towards its goal. How biological species map their environment is still an open area of research [32, 43]. In robotics, spatial representations have taken diverse forms, for instance metric maps [7, 14, 16] or topological maps [37, 39], allocentric or egocentric. Most of these variants have lately been presented in neural variants — metric neural maps [4, 19, 21, 30] or neural topological maps [5, 10, 34] learned from RL or with supervision.

In this work, we explore the question whether the emergence of mapping and spatial reasoning capabilities can be favored by the use of spatial auxiliary tasks that are related to a downstream objective. We target the problem of *Multi-Object Navigation*, where an agent must reach a sequence of specified objects in a particular order within a previously unknown

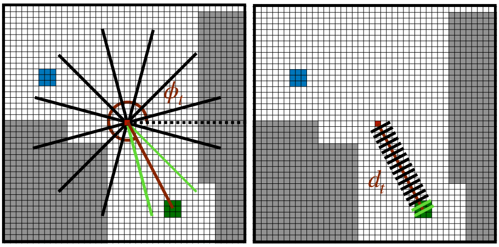

Figure 1: In the context of Deep-RL for Multi-Object Navigation, two auxiliary tasks predict the direction (*left*) and the distance (*right*) to the next object to retrieve *if it has been observed during the episode*. The **green object** is the current target. Both, the angle ϕ_t and the distance d_t between the center of the map (i.e. the agent) and the target at time t are discretized and associated with a class label.

environment. Such a task is interesting because it requires an agent to recall the position of previously encountered objects it will have to reach later in the sequence.

We take inspiration from the methodology in behavioral studies of human spatial navigation [15]. Experiments with human subjects aim at evaluating the spatial knowledge they acquire when navigating a given environment. In [15], two important measures are referred as the *sense of direction* and *judgement of relative distance*. Regarding knowledge of direction, a well-known task is *scene- and orientation- dependent pointing* (SOP), where participants must point to a specified location that is not currently within their field of view. Being able to assess its relative position compared to other objects in the world is critical to navigate properly, and disorientation is considered a main issue. In addition to direction, evaluating the distance to landmarks is also of high importance.

We conjecture, that an agent able to estimate the location of target objects relative to its current pose will implicitly extract more useful representations of the environment and navigate more efficiently. Classical methods based on RL rely on the capacity of the learning algorithm to extract mapping strategies from reward alone. While this has been shown to be possible in principle [4], we will show that the emergence of a spatial mapping strategy is significantly boosted through auxiliary tasks, which require the agent to continuously reason on the presence of targets w.r.t. to its viewpoint — see Figure 1.

To this end, we propose two auxiliary tasks, namely estimating the relative direction and the Euclidean distance to the current target object, conditioned on whether it has already been observed since the beginning of the episode. If an object is visible in the current observation, it will be helpful for training the agent to recognize it (discover its affordance) and estimate its relative position; spatial memory will be built up when the target was seen in the past.

We propose the following contributions: (i) we show that our proposed auxiliary tasks improve the performance of previous baselines by a large margin, which even allows to reach the performance of (incomparable) agents using ground-truth oracle maps as input; (ii) we show the consistency of the gains over different agents with multiple inductive biases, reaching from simple recurrent models to agents structured with projective geometry. This raises the question whether spatial inductive biases are required or whether spatial organization can be learned; (iii) the proposed method reaches SOTA performance on the *Multi-ON* task, and corresponds to the winning entry of the *CVPR 2021 Multi-ON challenge*.

2 Related Work

Visual navigation — has thus been extensively studied in robotics [6, 40]. An agent is placed in an unknown environment and must solve a specified task based on visual input, where [6] distinguish map-based and map-less navigation. Recently, many navigation problems have been posed as goal-reaching tasks [1]. The nature of the goal, it's regularities in the environment and how it is communicated to the agent have a significant impact on required reasoning capacities of the agent [3]. In *Pointgoal* [1], an agent must reach a location specified as relative coordinates, while *ObjectGoal* [1] requires the agent to find an object of a particular semantic category. Recent literature [3, 42] introduced new navigation tasks with two important characteristics, *(i)* their sequential nature, i.e. an episode is composed of a sequence of goals to reach, and *(ii)* the use of external objects as target objectives. *Multi-Object Navigation (MultiON)* [42] is a task requiring to sequentially retrieve objects, but unlike the *Ordered K-item* task [3], the order is not fixed between episodes. A sequential task is interesting as it requires the agent to remember and to map potential objects it might have seen while exploring the environment, as reasoning on them might be required in a later stage. Moreover, using external objects as goals prevents the agent from leveraging knowledge about the environment layouts, thus focusing solely on memory. Exploration is another targeted capacity as objects are placed randomly within environments. For all these reasons, our work thus focuses on the new challenging *Multi-ON* task [42].

Learning-free navigation — A recurrent pattern in methods tackling visual navigation [6, 40] is modularity, with different computational entities solving a particular sub-part of the problem. A module might map the environment, another one localize the agent within this map, a third one performing planning. Low-level control is also often addressed by a specialized sub-module. Known examples are based on Simultaneous Localization and Mapping (SLAM) [7, 14].

Learning-based navigation — The task of navigation can be framed as a learning problem, leveraging the abilities of deep networks to extract regularities from a large amount of training data. Formalisms range from Deep Reinforcement Learning (DRL) [23, 27, 49] to (supervised) Imitation Learning [12, 44, 46]. Such agents can be reactive [13, 49], but recent work tends to augment agents with memory, which is a key component, in particular in partially-observable environments [20, 29]. It can take the form of recurrent units [11, 22], or become a dedicated part of the system as in [18]. In the context of navigation, memory can full-fill multiple roles: holding a latent map-like representation of the spatial properties of the environment, as well as general high-level information related to the task ("*did I already see this object?*"). Common representations are metric [4, 19, 21, 30], or topological [5, 10, 34]. Other work reduces inductive biases by using Transformers [41] as a memory mechanism on episodic data [17, 33].

In contrast to end-to-end training, engineering stack approaches decompose the learning pipeline into sub-modules [9, 10] trained simultaneously with supervised learning [10] or a combination of supervised, reinforcement and imitation learning [9]. Somewhat related to our work, in [10], a dedicated semantic score prediction module is proposed, which estimates the direction towards a goal and is explicitly used to decide which previously unexplored ghost node to visit next inside a topological memory. In contrast, in oour work we propose to predict spatial metrics such as relative direction as an auxiliary objective to shape the learnt representations, instead of explicitly using those predictions at inference time.

Learning vs. learning-free — The differences in navigation performance between SLAM-based and learning-based agents has been studied before [25, 28, 35]. Even though trained agents begin to perform better than classical methods in recent studies, arguments regarding efficiency of SLAM-based methods in still hold [25, 28]. Frequently hybrid methods are suggested [9, 10]. In contrast, we explore the question, whether mapping strategies can emerge naturally in end-to-end training through additional pretext tasks.

Auxiliary tasks — can be combined with any downstream objective to guide a learning model to extract more useful representations as proposed in [23, 27] to improve, both, data efficiency and overall performance. [27] predict loop closure and reconstruct depth observations; Lample et al. [26] also augment the DRQN model [20] with predictions of game features in first-person shooter games. A potential drawback is the need for privileged information, which, however, is readily available in simulated environments [2, 24]. This is also the case in our work, where we access information during training on explored areas, positions of objects and of the agent, which, of course, is also used for reward generation in classical RL methods.

In [23], unsupervised objectives are introduced, such as pixel or action features and reward prediction. [47] introduce self-supervised auxiliary tasks to speed up the training on *PointGoal*. They augment the base agent from [45] with an inverse dynamics estimator as in [31], a temporal distance predictor, and an action-conditional contrastive module, which must differentiate between positives, i.e. real observations that occur after the given sequence, and negatives, i.e. observations sampled from other timesteps. [48] introduce auxiliary tasks for *ObjectGoal*, building on top of [47] and introduce the action distribution prediction and generalized inverse dynamics tasks and coverage prediction.

Our work belongs to the group of supervised auxiliary tasks, with an application to 3D complex and photo-realistic environments, which was not the case of concurrent methods. We also specifically target the learning of mapping and spatial reasoning through additional supervision, which has not been the scope of previous approaches.

3 Learning to map

We target the *Multi-ON* task [42], where an agent is required to reach a sequence of target objects in a certain order, and which was used for a recent challenge organized in the context of the CVPR 2021 Embodied AI Workshop. Compared to much easier tasks like *PointGoal* or (Single) *Object Navigation*, *Multi-ON* requires more difficult reasoning capacities, in particular mapping the position of an object once it has been seen. The following capacities are necessary to ensure optimal performance: (i) mapping the object, i.e. storing it in a suitable latent memory representation; (ii) retrieving this location on request and using it for navigation and planning. This also requires to decide when to retrieve this information, i.e. solving a correspondence problem between sub goals and memory representation.

The agent deals with sequences of objects that are randomly placed within the environment. At each time step, it only knows about the next object to find, which is updated when reached. The episode lasts until either the agent has found all objects in the correct order or the time limit is reached.

Inductive agent biases — Our contribution is independent of the actual inductive biases used for agents. We therefore explored different baseline agents with different architectures, as selected in [42]. The considered agents share a common base shown in Fig. 2, which extracts information from the current RGB-D observation. Variants also keep a global map that is first transformed into an egocentric representation centered around the agent's position, and possibly embeddings of the target object class and the previous actions. The vector

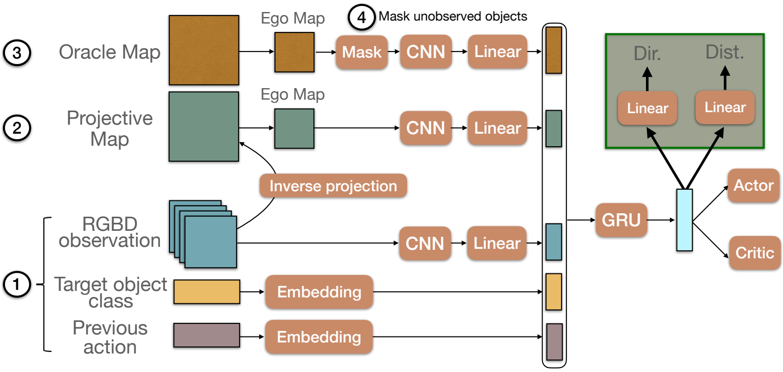

Figure 2: To study the impact of our auxiliary losses on different agents [42], we explore different input and inductive biases. All variants share basic observations ① (RGBD image, target class, previous action). Variants also use a map ② produced with inverse projective mapping as in [4, 21]. Oracle variants receive ground truth maps ③, where in one further variant unseen objects are removed ④. These architectures have been augmented with classification heads implementing the proposed auxiliary tasks (**green rectangle**).

representations are concatenated and fed to a GRU [11] unit that integrates temporal information, and whose output serves as input to an actor and a critic heads. These two modules respectively predict a distribution $\pi_\theta(a_t \mid s_t)$ over actions a_t conditioned on the current state s_t and the state-value function $V^{\pi_\theta}(s_t) = \mathbb{E}_{a_t \sim \pi_\theta}\left[\sum_{t'=t}^{T} \gamma^{t'} R_{t'} \mid S_t = s_t\right]$, i.e. expected cumulative reward starting in the current state s_t and following policy π_θ. The Actor-Critic algorithm is a baseline RL approach [38]. We consider different variants which have been explored in [42], but which have been introduced in prior work (numbers ①②③④ correspond to choices in Figure 2):

NoMap ① — is a recurrent GRU baseline without any spatial inductive bias.

ProjNeuralMap ①② **[4, 21]** — is a neural network structured with spatial information and projective geometry, in particular inverse 3D projection of the observed image features using a calibrated camera and depth information. Note that the notion of a “map” in this model refers to a network structure only, i.e. the map puts constraints on how input pixels are mapped to feature cells.

OracleMap ①③ — has access to a ground-truth grid map of the environment with 16 channels dedicated to occupancy information and 16 others to the presence of objects and their classes. As shown in Fig. 2, the map is cropped and centered around the agent to produce an egocentric map as input to the model.

OracleEgoMap ①③④ — gets the same egocentric map as OracleMap with only object channels, and revealed in regions that have already been within its field of view since the beginning of the episode. This variant corresponds to an agent capable of perfect mapping — no information gets lost, but only observed information is used.

3.1 Learning to map objects with auxiliary tasks

We introduce auxiliary tasks, additional to the classical RL objectives, and formulated as classification problems, which require the agent to predict information on object affordances, which were in its observation history in the current episode. To this end, the base model is augmented with two classification heads (Fig. 2) taking as input the contextual representation produced by the GRU unit:

Direction — the agent predicts the relative direction of the current target object, only if it has already been within the agent's field of view in the observation history of the current episode (Figure 1 left). The ground-truth direction towards the goal is first computed as follows,

$$\phi_t = \sphericalangle(\mathbf{o}_t, \mathbf{e}) = -\operatorname{atan2}(\mathbf{o}_{t,x} - \mathbf{e}_x, \mathbf{o}_{t,y} - \mathbf{e}_y) \tag{1}$$

where $\mathbf{e} = [\mathbf{e}_x \ \mathbf{e}_y]$ ("*ego*") are the coordinates of the agent on the grid and $\mathbf{o} = [\mathbf{o}_{t,x} \ \mathbf{o}_{t,x}]$ are the coordinates of the center of the target object at time t. As the ground-truth grid is egocentric, the position of the agent is fixed, i.e. at the center of the grid, while the target object gets different coordinates with time. The angles are kept in the interval $[0, 2\pi]$ and then discretized into K bins, giving the angle class. The ground-truth one-hot vector is denoted ϕ_t^*. At time instant t, the probability distribution over classes $\hat{\phi}_t$ is predicted from the GRU hidden state $\mathbf{h}_t$ through an MLP as $p(\hat{\phi}_t) = f_\phi(\mathbf{h_t}; \theta_\phi)$ with parameters θ_ϕ.

Distance — The second task requires the prediction of the Euclidean distance in the egocentric map between the center box, i.e. position of the agent, and the mean of the grid boxes containing the target object (Figure 1 right),

$$d_t = ||\mathbf{o}_t - \mathbf{e}||_2. \tag{2}$$

Again, distances are discretized into L bins, with d_t^* as ground-truth one-hot vector, and at time instant t, the probability distribution over classes $\hat{d}_t$ is predicted from the hidden state $\mathbf{h}_t$ through an MLP as $p(\hat{d}_t) = f_d(\mathbf{h_t}; \theta_d)$ with parameters θ_d.

Training — Following [42], all agents are trained with PPO [36] and a reward composed of three terms,

$$R_t = \mathbb{1}_{[\text{reached-goal}\ ]} \cdot R_{\text{goal}} + R_{\text{closer}} + R_{\text{time-penalty}} \tag{3}$$

where $\mathbb{1}_{[\text{reached-goal}\ ]}$ is the indicator function whose value is 1 if the *found* action was called while being close enough to the target, and 0 otherwise. R_{closer} is a reward shaping term equal to the decrease in geodesic distance to the next goal compared to previous timestep. Finally, $R_{\text{time-penalty}}$ is a negative slack reward to force the agent to take paths as short as possible.

PPO alternates between sampling and optimization phases. At sampling time k, a set $\mathcal{U}_k$ of trajectories τ with length T are collected using the latest policy, where T is smaller than the length of a full episode. The base PPO loss is then given as,

$$\mathcal{L}_{PPO} = \frac{1}{|\mathcal{U}_k|\,T} \sum_{\tau \in \mathcal{U}_k} \sum_{t=0}^{T-1} \left[\min\left(r_t(\theta)\hat{A}_t, \operatorname{clip}\left(r_t(\theta), 1-\varepsilon, 1+\varepsilon\right)\hat{A}_t\right)\right] \tag{4}$$

where $\hat{A}_t$ is an estimate of the advantage function $A^{\pi_\theta}(s_t, a_t) = Q^{\pi_\theta}(s_t, a_t) - V^{\pi_\theta}(s_t)$ at time t, and $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$ is the probability ratio between the updated and old versions of the policy. We did not make the dependency of states and actions on the trajectory τ explicit in the notation.

Both direction and distance predictions are supervised with cross-entropy losses from

Agent	**Dir.**	**Dist.**	**Success**	**Progress**	**SPL**	**PPL**	**Comparable**
OracleMap	–	–	50.9± 2.4	61.2± 2.0	40.7± 1.9	48.8± 1.4	–
OracleEgoMap	–	–	34.1± 4.1	48.8± 3.8	27.9± 2.5	39.6± 2.2	–
ProjNeuralMap	–	–	27.2± 2.9	44.6± 2.2	19.5± 0.7	32.3± 0.3	✓
	✓	–	45.9± 6.9	61.2± 5.1	32.8± 5.0	43.7± 3.8	✓
	✓	✓	**51.3 ± 7.8**	**65.5 ± 5.8**	**35.9 ± 5.3**	**46.0 ± 4.0**	✓

Table 1: Impact of different auxiliary tasks (validation performance). Results are reported after 4 training runs with different random seeds per model, without cherry-picking of best performing ones. Direction prediction significantly improves the performance of *ProjNeuralMap* baseline, adding distance prediction further increases the downstream performance by a large margin, matching the performance of (incomparable!) *OracleEgoMap*. Both losses are effective and complementary.

Agent	**Aux. Sup.**	**Success**	**Progress**	**SPL**	**PPL**	**Comparable**
OracleMap	–	41.9± 3.0	52.7± 3.7	32.9± 1.9	41.1± 2.4	–
OracleEgoMap	–	22.3± 4.5	37.1± 5.1	17.4± 3.0	28.7± 2.7	–
ProjNeuralMap	–	17.9± 1.6	34.5± 2.1	12.2± 0.6	23.9± 0.6	✓
	✓	**36.0 ± 6.0**	**51.2 ± 5.0**	**24.2 ± 3.9**	**34.9 ± 3.4**	✓
NoMap	–	8.3± 1.4	22.8± 1.5	6.4± 0.6	17.6± 0.5	✓
	✓	23.0± 2.6	39.2± 2.6	15.1± 2.0	26.2± 2.0	✓

Table 2: Consistency over multiple models (test set). Results are reported after 4 training runs with different random seeds per model, without cherry-picking of best performing ones. *ProjNeuralMap* performs significantly better when trained with our additional losses, even outperforming *OracleEgoMap*. The recurrent-only *NoMap* agent does not have any spatial inductive bias. When augmented with our supervision it outperforms *ProjNeuralMap*, also closing the gap with *OracleEgoMap*. This provides evidence that spatial inductive bias provides an edge, but that an unstructured agent can decrease the gap through supervision.

ground truth values ϕ^*_t and d^*_t, respectively, as

$$\mathcal{L}_\phi = \frac{1}{|\mathcal{U}_k|T}\sum_{\tau\in\mathcal{U}_k}\sum_{t=0}^{T-1}\left[-\mathbb{1}_t\sum_{c=1}^{K}\phi^*_{t,c}\log p(\hat{\phi}_{t,c})\right], \quad \mathcal{L}_d = \frac{1}{|\mathcal{U}_k|T}\sum_{\tau\in\mathcal{U}_k}\sum_{t=0}^{T-1}\left[-\mathbb{1}_t\sum_{c=1}^{L}d^*_{t,c}\log p(\hat{d}_{t,c})\right] \tag{5}$$

where $\mathbb{1}_t$ is the binary indicator function specifying whether the current target object has already been seen in the current episode ($\mathbb{1}_t$=1), or not ($\mathbb{1}_t$=0).

The auxiliary losses $\mathcal{L}_\phi$ and $\mathcal{L}_d$ are added as follows,

$$\mathcal{L}_{tot} = \mathcal{L}_{PPO} + \lambda_\phi \mathcal{L}_\phi + \lambda_d \mathcal{L}_d \tag{6}$$

where λ_ϕ and λ_d weight the relative importance of both auxiliary losses.

4 Experimental Results

We focus on the *3-ON* version of the Multi-ON task, where the agent deals with sequences of 3 objects. The time limit is fixed to 2500 environment steps, and there are 8 object classes. The agent receives a (256×256×4) RGB-D observation and the one-in-K encoded class of

Agent/Method	— Test Challenge —				— Test Standard —			
	Success	Progress	SPL	PPL	Success	Progress	SPL	PPL
Ours (Auxiliary losses)	**55**	**67**	**35**	**44**	57	70	36	45
Team 2	52	64	32	38	62	71	34	39
Team 3	41	57	26	36	43	57	27	36
ProjNeuralMap (Challenge baseline)	–	–	–	–	12	29	6	16
NoMap (Challenge baseline)	–	–	–	–	5	19	3	13

Table 3: Our method corresponds to the winning entry in the CVPR 2021 Multi-ON Challenge Leaderboard: *Test Challenge* are the official challenge results. *Test Standard* contains pre- and post-challenge results. The official challenge ranking is done with **PPL**, which evaluates correct mapping (quicker and more direct finding of objects).

the current target object within the sequence. The discrete action space is composed of four actions: *move forward* 0.25*m*, *turn left* 30°, *turn right* 30°, and *found*, which signals that the agent considers the current target object to be reached. As the aim of the task is to focus on evaluating the importance of mapping, a perfect localization of the agent was assumed as in the protocol proposed in [42].

Dataset and metrics — we used the standard train/val/test split over scenes from the Matterport [8] dataset. 1000 episodes are sampled from the val and test splits for model validation and testing, respectively. We consider standard metrics of the field as given in [42]:

- *Success*: percentage of successful episodes (the agent reaches all the three objects in the right order in the time limit).
- *Progress*: percentage of objects successfully found in an episode.
- *SPL*: Success weighted by Path Length. This extends the original SPL metrics from [1] to the sequential multi-object case.
- *PPL*: Progress weighted By Path Length.

Note that for an object to be considered found, the agent must take the *found* action while being within 1.5m of the current goal. The episode ends immediately if the agent calls *found* in an incorrect location. For more details, we refer to [42].

Implementation details — training and evaluation hyper-parameters, as well as architecture details have been taken from [42]. All reported quantitative results are obtained after 4 training runs for each model, during 70*M* steps (increased from 40*M* in [42]). Ground-truth direction and distance measures are respectively split into $K = 12$ and $L = 36$ classes. Indeed, angle bins span 30°, and distance bins span a unit distance on the egocentric map, that is 50*x*50 (the maximum distance between center and a grid corner is thus 35). Training weights λ_ϕ and λ_d are both fixed to 0.25.

Do the auxiliary tasks improve the downstream objective? — in Table 1, we study the impact of both auxiliary tasks on the 3-ON benchmark when added to the training objective of *ProjNeuralMap*, and their complementarity. Direction prediction significantly improves performance, adding distance prediction further increases the downstream performance by a large margin. Both losses have thus a strong impact and are complementary, confirming the assumption that *sense of direction* and *judgement of relative distance* are two key skills for spatially navigating agents.

Table 2 presents results on the test set, confirming the significant impact on each of the considered metrics. *ProjNeuralMap* with auxiliary losses matches the performance of (incomparable!) *OracleMap* on Progress. *OracleMap* has higher PPL and SPL, but has also

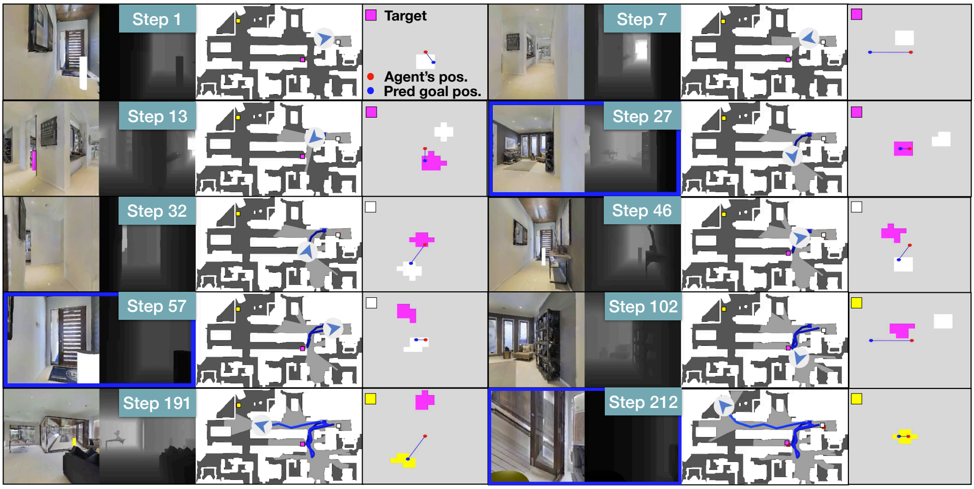

Figure 3: Example agent trajectory (sample from competition Mini-val set). The agent properly explores the environment to find the pink object. It then successfully backtracks to reach the white cylinder, and finally goes to the yellow one after another exploration phase (see text for a detailed description). The relative direction and distance predictions are combined into a visualised blue point on top of the oracle egocentric map. Note that these predictions are not used by the agent at inference time, and are only shown for visualisation purposes. The top down view and oracle egocentric map are also provided for visualisation only.

access to very strong privileged information.

Can an unstructured recurrent agent learn to map? — we explore whether an agent without spatial inductive bias can be trained to learn a mapping strategy, to encode spatial properties of the environment into its unstructured hidden representation. As shown in Table 2, *NoMap* indeed strongly benefits from the auxiliary supervision (Success for instance jumping from 7.4% to 22.4%). Improvement is significant, closing the gap with *ProjNeuralMap* trained with vanilla RL. The quality of extra supervision can thus help to guide the learnt representation, mitigating the need for incorporating inductive biases into neural networks. When both are trained with our auxiliary losses, *ProjNeural* still outperforms *NoMap*, indicating that spatial inductive bias still provides an edge.

Comparison with the state-of-the-art — our method corresponds to the winning entry of the *CVPR 2021 Multi-On Challenge* organized with the *Embodied AI Workshop*, shown in Table 3. Compared to the method described above, the challenge entry contained a third additional auxiliary loss, which required the agent to predict whether an object had been seen or not in the observation history. Post-challenge analysis however showed, that this third loss did not have an impact. The official challenge ranking is done with **PPL**, which evaluates correct mapping (quicker and more direct finding of objects), while mapping does not necessarily have an impact on success rate, which can be obtained by pure exploration.

Visualization — Fig. 3 illustrates an example trajectory from the agent trained with the auxiliary supervision in the context of the *CVPR 2021 Multi-On Challenge*. The agent starts the episode (Step 1) seeing the white object, which is not the first target to reach. It thus starts exploring the environment (Step 7), until seeing the pink target object (Step 13). Its prediction of the goal distance immediately improves, showing it is able to recognize the object within the RGB-D input. The agent then reaches the target (Step 27). The new target is now the white object (that was seen in Step 1). While it is still not within its current filed of view, the agent can localize it quite precisely (Step 32), and go towards the goal (Step 46) to call the *found* action (Step 57). The agent must then explore again to find the last object (Step 105). When the yellow cylinder is seen, the agent can estimate its relative position (Step 191) before reaching it (Step 212) and ending the episode.

5 Conclusion

In this work, we propose to guide the learning of mapping and spatial reasoning capabilities by augmenting vanilla RL training objectives with auxiliary tasks. We show that learning to predict the relative direction and distance of already seen target objects improves significantly the performance on various metrics and that these gains are consistent over agents with or without spatial inductive bias. We reach SOTA performance on the Multi-ON benchmark. Future work will investigate additional structure, for instance predicting multiple objects.

References

[1] Peter Anderson, Angel X. Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, and Amir Roshan Zamir. On evaluation of embodied navigation agents. *arXiv preprint*, 2018.

[2] Charles Beattie, Joel Z Leibo, Denis Teplyashin, Tom Ward, Marcus Wainwright, Heinrich Küttler, Andrew Lefrancq, Simon Green, Víctor Valdés, Amir Sadik, et al. Deepmind lab. *arXiv preprint*, 2016.

[3] Edward Beeching, Jilles Dibangoye, Olivier Simonin, and Christian Wolf. Deep reinforcement learning on a budget: 3d control and reasoning without a supercomputer. In *International Conference on Pattern Recognition*, 2020.

[4] Edward Beeching, Jilles Dibangoye, Olivier Simonin, and Christian Wolf. Egomap: Projective mapping and structured egocentric memory for deep RL. In *ECML-PKDD*, 2020.

[5] Edward Beeching, Jilles Dibangoye, Olivier Simonin, and Christian Wolf. Learning to plan with uncertain topological maps. In *European Conference on Computer Vision 2020*, 2020.

[6] Francisco Bonin-Font, Alberto Ortiz, and Gabriel Oliver. Visual navigation for mobile robots: A survey. *Journal of intelligent and robotic systems*, 2008.

[7] Guillaume Bresson, Zayed Alsayed, Li Yu, and Sébastien Glaser. Simultaneous localization and mapping: A survey of current trends in autonomous driving. *IEEE Transactions on Intelligent Vehicles*, 2017.

[8] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niebner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. In *IEEE International Conference on 3D Vision*, 2018.

[9] Devendra Singh Chaplot, Dhiraj Gandhi, Saurabh Gupta, Abhinav Gupta, and Ruslan Salakhutdinov. Learning to explore using active neural slam. In *International Conference on Learning Representations*, 2020.

[10] Devendra Singh Chaplot, Ruslan Salakhutdinov, Abhinav Gupta, and Saurabh Gupta. Neural topological slam for visual navigation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[11] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing*, 2014.

[12] Yiming Ding, Carlos Florensa, Pieter Abbeel, and Mariano Phielipp. Goal-conditioned imitation learning. In *Advances in Neural Information Processing Systems*, 2019.

[13] Alexey Dosovitskiy and Vladlen Koltun. Learning to act by predicting the future. In *International Conference on Learning Representations*, 2017.

[14] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine*, 2006.

[15] Arne D Ekstrom, Hugo J Spiers, Véronique D Bohbot, and R Shayna Rosenbaum. *Human spatial navigation*. Princeton University Press, 2018.

[16] Alberto Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer*, 1989.

[17] Kuan Fang, Alexander Toshev, Li Fei-Fei, and Silvio Savarese. Scene memory transformer for embodied agents in long-horizon tasks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[18] Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *arXiv preprint*, 2014.

[19] Saurabh Gupta, James Davidson, Sergey Levine, Rahul Sukthankar, and Jitendra Malik. Cognitive mapping and planning for visual navigation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2616–2625, 2017.

[20] Matthew Hausknecht and Peter Stone. Deep recurrent q-learning for partially observable mdps. In *AAAI Conference on Artificial Intelligence*, 2015.

[21] João F. Henriques and Andrea Vedaldi. Mapnet: An allocentric spatial memory for mapping environments. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[22] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 1997.

[23] Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z. Leibo, David Silver, and Koray Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. In *International Conference on Learning Representations*, 2017.

[24] Michał Kempka, Marek Wydmuch, Grzegorz Runc, Jakub Toczek, and Wojciech Jaśkowski. Vizdoom: A doom-based ai research platform for visual reinforcement learning. In *IEEE Conference on Computational Intelligence and Games*, 2016.

[25] Noriyuki Kojima and Jia Deng. To learn or not to learn: Analyzing the role of learning for navigation in virtual environments. *arXiv preprint*, 2019.

[26] Guillaume Lample and Devendra Singh Chaplot. Playing fps games with deep reinforcement learning. In *AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[27] Piotr Mirowski, Razvan Pascanu, Fabio Viola, Hubert Soyer, Andy Ballard, Andrea Banino, Misha Denil, Ross Goroshin, Laurent Sifre, Koray Kavukcuoglu, Dharshan Kumaran, and Raia Hadsell. Learning to navigate in complex environments. In *International Conference on Learning Representations*, 2017.

[28] Dmytro Mishkin, Alexey Dosovitskiy, and Vladlen Koltun. Benchmarking classic and learned navigation in complex 3d environments. *arXiv preprint*, 2019.

[29] Junhyuk Oh, Valliappa Chockalingam, Satinder, and Honglak Lee. Control of memory, active perception, and action in minecraft. In *International Conference on Machine Learning*, 2016.

[30] Emilio Parisotto and Ruslan Salakhutdinov. Neural map: Structured memory for deep reinforcement learning. In *International Conference on Learning Representations*, 2018.

[31] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning*, 2017.

[32] Michael Peer, Iva K Brunec, Nora S Newcombe, and Russell A Epstein. Structuring knowledge with cognitive maps and cognitive graphs. *Trends in Cognitive Sciences*, 2020.

[33] Samuel Ritter, Ryan Faulkner, Laurent Sartran, Adam Santoro, Matthew Botvinick, and David Raposo. Rapid task-solving in novel environments. In *International Conference on Learning Representations*, 2021.

[34] Nikolay Savinov, Alexey Dosovitskiy, and Vladlen Koltun. Semi-parametric topological memory for navigation. In *International Conference on Learning Representations*, 2018.

[35] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A platform for embodied ai research. In *IEEE/CVF International Conference on Computer Vision*, 2019.

[36] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint*, 2017.

[37] Hagit Shatkay and Leslie Pack Kaelbling. Learning topological maps with weak local odometric information. In *IJCAI*, 1997.

[38] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[39] Sebastian Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 1998.

[40] Sebastian Thrun, Wolfram Burgard, Dieter Fox, et al. Probabilistic robotics, vol. 1, 2005.

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

[42] Saim Wani, Shivansh Patel, Unnat Jain, Angel X. Chang, and Manolis Savva. Multion: Benchmarking semantic map memory using multi-object navigation. In *Advances in Neural Information Processing Systems*, 2020.

[43] William H Warren, Daniel B Rothman, Benjamin H Schnapp, and Jonathan D Ericson. Wormholes in virtual space: From cognitive maps to cognitive graphs. *Cognition*, 2017.

[44] David Watkins-Valls, Jingxi Xu, Nicholas Waytowich, and Peter Allen. Learning your way without map or compass: Panoramic target driven visual navigation. *arXiv preprint*, 2019.

[45] Erik Wijmans, Abhishek Kadian, Ari Morcos, Stefan Lee, Irfan Essa, Devi Parikh, Manolis Savva, and Dhruv Batra. Dd-ppo: Learning near-perfect pointgoal navigators from 2.5 billion frames. In *International Conference on Learning Representations*, 2019.

[46] Qiaoyun Wu, Xiaoxi Gong, Kai Xu, Dinesh Manocha, Jingxuan Dong, and Jun Wang. Towards target-driven visual navigation in indoor scenes via generative imitation learning. *arXiv preprint*, 2020.

[47] Joel Ye, Dhruv Batra, Erik Wijmans, and Abhishek Das. Auxiliary tasks speed up learning pointgoal navigation. *arXiv preprint*, 2020.

[48] Joel Ye, Dhruv Batra, Abhishek Das, and Erik Wijmans. Auxiliary tasks and exploration enable objectnav. *arXiv preprint*, 2021.

[49] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J. Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *International Conference on Robotics and Automation*, 2017.