Cross-View Policy Learning for Street Navigation

Ang Li[1]* Huiyi Hu[1]* Piotr Mirowski[2] Mehrdad Farajtabar[1]
[1]DeepMind, Mountain View, CA
[2]DeepMind, London, UK
{anglili, clarahu, piotrmirowski, farajtabar}@google.com

Abstract

The ability to navigate from visual observations in unfamiliar environments is a core component of intelligent agents and an ongoing challenge for Deep Reinforcement Learning (RL). Street View can be a sensible testbed for such RL agents, because it provides real-world photographic imagery at ground level, with diverse street appearances; it has been made into an interactive environment called StreetLearn [27] and used for research on navigation. However, goal-driven street navigation agents have not so far been able to transfer to unseen areas without extensive retraining, and relying on simulation is not a scalable solution. Since aerial images are easily and globally accessible, we propose instead to train a multi-modal policy on ground and aerial views, then transfer the ground view policy to unseen (target) parts of the city by utilizing aerial view observations. Our core idea is to pair the ground view with an aerial view and to learn a joint policy that is transferable across views. We achieve this by learning a similar embedding space for both views, distilling the policy across views and dropping out visual modalities. We further reformulate the transfer learning paradigm into three stages: 1) cross-modal training, when the agent is initially trained on multiple city regions, 2) aerial view-only adaptation to a new area, when the agent is adapted to a held-out region using only the easily obtainable aerial view, and 3) ground view-only transfer, when the agent is tested on navigation tasks on unseen ground views, without aerial imagery. Experimental results suggest that the proposed cross-view policy learning enables better generalization of the agent and allows for more effective transfer to unseen environments.

1. Introduction

Stranded on Elephant Island after the shipwreck of the *Endurance* expedition, Ernest Shackleton, Frank Worsley and their crew attempted, on 24 April 1916, a risky 720-mile open-boat journey to South Georgia. They had duly studied the trajectory using nautical maps, but the latter froze and became illegible. It is only through their extraordinary navigation skills, memory, and by transferring knowledge derived from a top-view representation to visual and compass observations as they sailed, that they ultimately reached the shores of South Georgia two weeks later. Such a feat has been cited as a prime example of complex human spatial navigation in unknown environments [10]: having gained expertise in navigating using both maps and sea-level observations, they could adapt to an unknown environment by studying maps and then transfer that knowledge on their new journey.

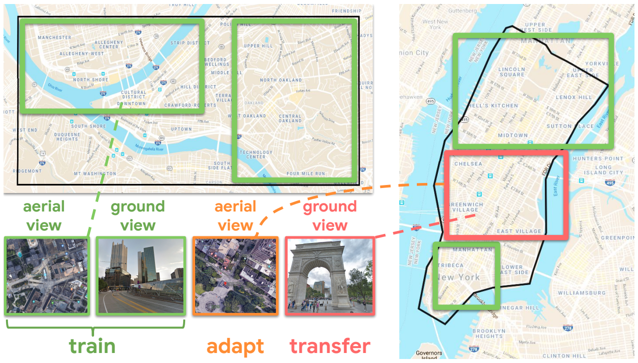


Figure 1. The street navigation agent observes both ground and aerial views in the training phase. The agent learns a view-invariant policy to associate the two views. Once the policy is learned, the agent becomes capable of continual training with interchangeable viewpoints. When being transferred to an unseen area, the agent is adapted using only the aerial view observations, which are easily accessible. The agent is then transferred to the ground view environment (without access to aerial-view images) for testing. Images: Google Maps and Street View.

The ability to navigate in familiar and unfamiliar environments is a core component of animal and artificial intelligence. The research on artificial agent navigation can be applied to real world domains ranging from the neuroscience of grid and place cells in mammals [3, 9] to the autonomy of indoor and outdoor mobile robots [49, 39, 31, 38, 45, 32].

[*] Equal contribution

We focus on the visual navigation task that trains an agent to navigate in a specific area by using a single sensory modality, integrates visual perception and decision making processes, and typically does not rely on maps. A challenging question arises: *how to efficiently transfer the agents to new or previously unseen areas?* In the absence of extra information, existing solutions typically require to retrain the agent on that unseen area, which is computationally expensive [6]. Alternatively, one can simplify navigation tasks so as not to require local knowledge [49] or to rely on additional navigation instructions [8, 15]. Generalization to unseen environments can be obtained by approaching navigation as a one-shot learning task with an auxiliary memory in simple and procedurally generated environments [43, 47] or by building complex simulators for more complex environments [31, 39]. It is however expensive to build a simulator for offline retraining (especially in the case of unconstrained outdoor environments) and street-level images are expensive to collect as one has to drive everywhere to take panoramic photographs. As a consequence, enabling an agent to navigate in unseen locations, without fully retraining it from scratch, is still a challenging problem.

Inspired by the observation that humans can quickly adapt to a new city simply by reading a map, we explore the idea of incorporating comparable top-down visual information into the training procedure of navigation agents, in order to help them generalize to previously unseen streets. Instead of using a human-drawn map, we choose aerial imagery, as it is readily available around the world. Moreover, humans can easily do without maps once they become familiar with an environment. This human versatility motivates our work on training flexible RL agents that can perform using both first-person and top-down views.

We propose a novel solution to improve transfer learning for visual navigation in cities, leveraging easily accessible aerial images (Figure 1). These aerial images are collected for both source (training) and target (unseen or held-out) regions and they are paired with ground-level (street-level or first-person) views based on their geographical coordinates. We decompose the transfer task into three stages: *training* on both ground-view and aerial-view observations in the source regions, *adaptation* using only the aerial-view observations in the target region, and *transfer* of the agent to the target area using only ground-view observations. Note that our goal remains to train agents to navigate from ground-view observations. The RL agent should therefore have access to the aerial views only during the first (training) and second (adaptation) stages, but not during the third (transfer) stage when it is deployed in the target area.

The gist of our solution is transfering the agent to an unseen area using an auxiliary environment built upon a different but easily accessible modality – the aerial images. This requires the agent to be flexible at training time by relying on interchangeable observations. We propose a cross-view framework to learn a policy that is invariant to different viewpoints (ground view and aerial view). Learning view-invariant policy relies on three main ingredients: (a) an L_2 distance loss to minimize the embedding distance between the two views, (b) a dual pathway, each with its own policy, with a Kullback-Leibler (KL) loss on the policy logits to force these two policies to be similar, and (c) a dropout module called *view dropout* that randomly chooses the policy logits from either view to select actions. The proposed architecture naturally works with interchangeable observations and is flexible for training with both views jointly or with only view at a time. This makes it a flexible model that can be shared across the three stages of transfer learning.

We build our cross-view policy architecture by extending the RL agents proposed in [29] into a two-stream model that corresponds to the two views. Our agents are composed of three modules: a convolutional network [22] responsible for visual perception, a local recurrent neural network (RNN) or Long Short-Term Memory (LSTM) [17] for capturing location-specific features (*locale* LSTM), and a policy RNN producing a distribution over the actions (*policy* LSTM).

We build our testbed, called *StreetAir* (to the best of our knowledge, the first multi-view outdoor street environment), on top of *StreetLearn*, an interactive first-person street environment built upon panoramic street-view photographs [27]. We evaluate it on the same task as in [29], namely goal driven navigation or the *courier* task, where the agent is only given the latitude and longitude coordinates of a goal destination, without ever being given its current position, and learns to both localize itself and plan a trajectory to the destination. Our results suggest that the proposed method transfers agents to unseen regions with higher zero-shot rewards (transfer without training in the held-out ground-view environment) and better overall performance (continuously trained during transfer) compared to single-view (ground-view) agents.

Contribution. Our contributions are as follows.

1. We propose to transfer the ground-view navigation task between areas by leveraging a paired environment based upon easily accessible aerial-view images.
2. We propose a cross-view policy learning framework to encourage transfer between observation modalities via both representation-level and policy-level associations, and a novel view dropout to force the agent to be flexible and to use ground and aerial views interchangeably.
3. We propose a three-stage procedure as a general recipe for transfer learning: cross-modal training, adaptation using auxiliary modality, and transfer on main modality.
4. We implement and evaluate our agents on *StreetAir*, a realistic multi-view street navigation environment that extends *StreetLearn* [27].

2. Related Work

2.1. Visual Navigation

Zhu *et al*. [49] proposed an actor-critic model whose policy was a function of the goal as well as of the current state, both presented as images. Subsequent work on Deep Reinforcement Learning focused on implicit goal-driven visual navigation [28, 7, 39, 46] and addressed generalization in unseen environments through implicit [33, 43] or explicit [47, 36] map representations. Gupta *et al*. [13] introduced landmark- and map-based navigation using a spatial representation for path planning and a goal-driven closed-loop controller for executing the plan. A successor-feature-based deep RL algorithm that can learn to transfer knowledge from previously mastered navigation tasks to new problem instances was proposed in [46]. However, the above works either relied on simulators or attained navigation in simple, unrealistic, or limited indoor environments.

There has been a growing interest in building and benchmarking visual navigation using complex simulators [21, 38] or photorealistic indoor environments [31]. By contrast, we built our work on the top of a realistic environment *StreetLearn* [29, 27], made from Google Street View imagery and Google Maps street connectivity.

2.2. Cross-View Matching

Matching street viewpoints with aerial imagery has been a challenging computer vision problem [23, 20, 25, 34]. Recent approaches include geometry-based methods and deep learning. Li *et al*. would extract geometric structures on the ground between street and ortho view images, and measure the similarity between modalities by matching their linear structures [23]. Bansal *et al*. [4] proposed to match lines on the building facades. Lin *et al*. proposed to learn a joint embedding space using a deep neural network between street views and aerial views [25, 41]. All these works aim at utilizing cross-view matching to achieve image-based geolocation - specifically, finding the nearest neighbors, in some embedding space, between the query street image and all the geo-referenced aerial images in the database. Our work is closely related to cross-view matching, but instead of supervised learning, we study how cross-view learning could improve RL-based navigation tasks.

2.3. Multimodal Learning

Our work is also generally related to multimodal learning since street views and aerial views are not taken from the same type of cameras; they are basically from two different modalities. Many of the existing multimodal learning works focus on merging language and visual information. In the visual navigation domain, Hermann *et al*. built upon the *StreetLearn* environment [29] with additional inputs from language instructions, to train agents to navigate in a city by following textual directions [15]. Anderson *et al*. proposed the vision-and-language navigation (VLN) task based upon an indoor environment [1]. Wang *et al*. [42] proposed to learn, from paired trajectories and instructions, a cross-modal critic that provides intrinsic rewards to the policy and utilizes self-supervised imitation learning.

2.4. Knowledge Distillation

Our work is related to Network Distillation [16, 2] and its many extensions [30, 24, 48, 37], as one way to transfer knowledge. A *student* network tries to indirectly learn from a *teacher* network by imposing a Kullback-Leibler (KL) loss between its own and the teacher's softened logits, *i.e.*, trying to mimic the teacher's behavior. In [14] Gupta *et al*. generalize knowledge distillation for two modalities (RGB and depth) at the final layer by minimizing the L_2 loss for object and action detection. The hallucination network in [18] was trained on an existing modality to regress the missing modality using L_2 loss, and leveraged multiple such losses for multiple tasks. This work has been extended by Garcia *et al*. by adding L_2 losses for reconstructing all layers of the depth network and a cross entropy distillation loss for a missing network [12]. Finally, Luo *et al*. [26] learned the direction of distillation between modalities, considering a cosine distillation loss and a representation loss.

Our work differs in three ways: First, distillation has been applied to either classification or object/activity detection, while our work focuses on transferring knowledge in a control problem by distilling both image representations and RL policies. Second, distillation has so far been applied from a teacher network to a student network, while we choose to transfer between the auxiliary task (aerial view) and the main task (street view), sharing the local and policy modules in the network. Third, we employ a novel view dropout to further enhance the transferablity.

2.5. Transfer Learning

Our work is related to transfer learning [35] in visual domains. The very basic approach to transfer learning is to pretrain on an existing domain or task and fine-tune on the target ones. Luo *et al*. [26] proposed a method to transfer multimodal privileged information across domains for action detection and classification. Chaplot *et al*. [6] studied the effectiveness of pretraining and fine-tuning for transferring knowledge between various environments for 3D navigation. Kansky *et al*. [19] proposed Schema Networks to transfer experience from one scenario to other similar scenarios that exhibit repeatable structure and sub-structure. Bruce *et al*. [5] leverage an interactive world model built from a single traversal of the environment, a pretrained visual feature encoder, and stochastic environmental augmentation, to demonstrate successful transfer under real-world environmental variations without fine-tuning.

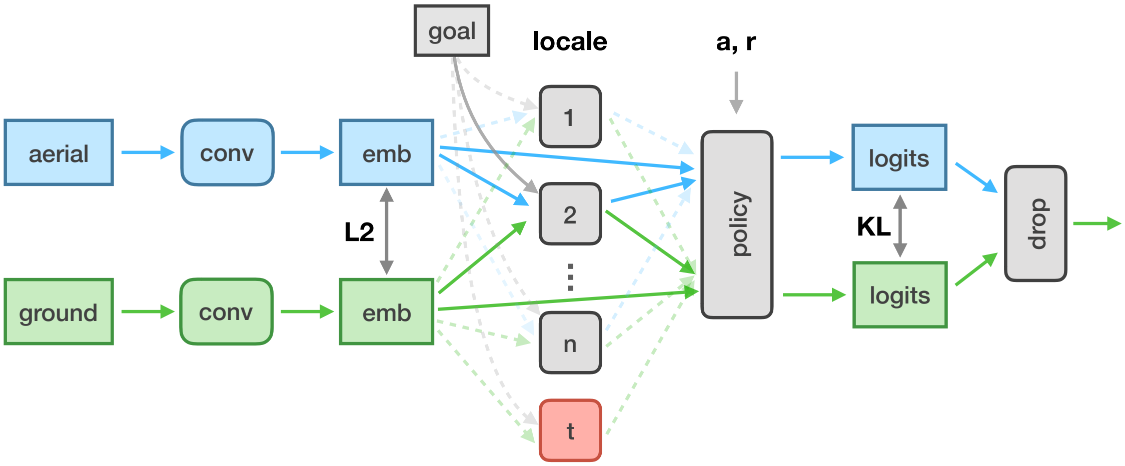


Figure 2. Overview of Cross-view Policy Learning: Ground-view and aerial-view inputs are passed into separate Convolutional Neural Networks for embedding. An L_2 embedding loss is used to constrain the similarity between the two latent spaces. The embeddings are passed to a *locale* LSTM (region-specific) and a global policy LSTM (shared across all regions). Both LSTMs are shared across the two views. A KL policy loss is used to constrain the policy logits between the two views. View dropout (gating) selects either of the two views and the final action is sampled according to a multinomial distribution over the logits. This figure shows n regions (gray boxes) for training and one target region (red box) for transfer. Goals are represented by lat/long coordinates. a, r represent the action and reward respectively.

3. Approach: Cross-view Policy Learning

The full model of our navigation agent is illustrated on Figure 2. Both ground-level and aerial view images are fed into the corresponding representation networks, Convolutional Neural Networks (CNN) [22] without weight sharing across the two modalities. The image embeddings, output by the CNNs, are then passed into a *locale*-specific LSTM, whose output is then fed into the *policy* LSTM together with the visual embedding. The *policy* LSTM produces logits of a multinomial distribution over actions. As there are two pathways (for ground-level and aerial views) with two sets of policy logits, an additional gating function decides the final set of logits (either by choosing or merging the two policies) from which to sample the action.

In order to bind the two views and to allow for learning a policy that is interchangeable across views, we proposed to incorporate three ingredients as part of this cross-view policy learning framework: an embedding loss, a policy distillation loss and view dropout, which we detail in the subsequent sections.

3.1. Reinforcement Learning

We follow [29] and employ the policy gradient method for training the navigation agents, learning a policy π that maximizes the expected reward $\mathbb{E}[\mathcal{R}]$. In this work, we use a variant of the REINFORCE [44] advantage actor-critic algorithm $\mathbb{E}_{a_t \sim \pi_\theta}[\sum_t \nabla_\theta \log \pi(a_t|s_t, \mathbf{g}; \theta)(\mathcal{R}_t - \mathcal{V}^\pi(s_t))]$, where $\mathcal{R}_t = \sum_{j=0}^{T-t} \gamma^j r_{t+j}$, r_t is the reward at time t, γ is a discounting factor, and T is the episode length. In this work, instead of representing the goal $\mathbf{g}$ using distances to pre-determined landmarks, we directly use latitude and longitude coordinates.

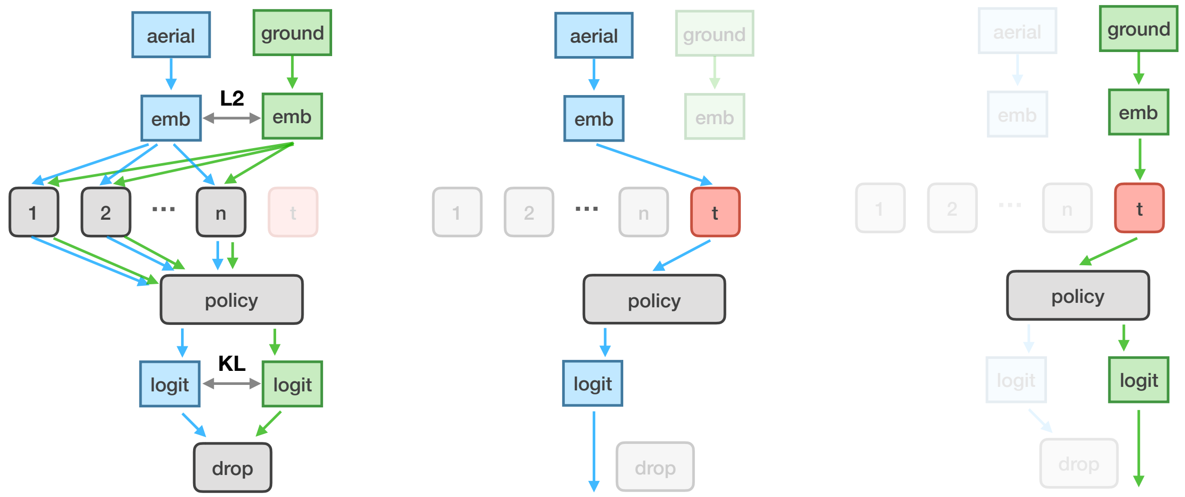
We specifically train the agents using IMPALA [11], a distributed asynchronous actor-critic implementation of RL, with 256 actors for single-region and 512 actors for multi-region experiments, relying on off-policy minibatches re-weighted by importance sampling. Curriculum learning and reward shaping are used in the early stage of agent training to smooth out the learning procedure, similarly to [29].

3.2. Joint Multi-View Embedding

There are two reasons why we need to learn a joint representation between the two views in order to exploit the auxiliary aerial view. First, learning a joint embedding enables us to substitute aerial views for ground-level views at transfer time, once we have adapted the agent to the unseen area using aerial views only. Secondly, enforcing the embeddings to be similar could potentially make model training faster and more robust. The original representation is only learned through interactions with the environment so ideally such representation should not be dissimilar when one uses signals from different modalities. Motivated by these, we introduce an embedding loss that enforces learning a joint embedding space between the two views:

$$\ell_{\text{embed}} = \|f_g(x_{\text{ground}}) - f_a(x_{\text{aerial}})\|_2 \,, \tag{1}$$

where f_g and f_a are the CNN modules corresponding to ground-level and aerial view inputs, respectively.



(a) Training (b) Adaptation (c) Transfer

Figure 3. Transfer learning procedure including 3 stages: training, adaptation and transfer. The agent is trained with both ground and aerial view observations in the training city regions. Part of the agent is adapted to the held-out city region, using only aerial view observations. The agent is transferred to the target city region and continuously trained using only ground-view observations.

3.3. Policy Distillation

Simply minimizing the L_2 distance between embeddings may not be sufficient since in practice it is impossible to exactly match one with the other. The small errors between the two representations could be amplified dramatically as they propagate into the policy networks. So we further propose to match the logits between the policy outputs from the two modalities. In other words, although the embedding between the two modalities may be slightly different, the policy should always try to generate the same actions at the end. Specifically, a Kullback-Leibler divergence loss is added to the total loss, *i.e.*,

$$\ell_{\text{policy}} = -\sum_{x} p_g(x) \log\left(\frac{p_a(x)}{p_g(x)}\right), \tag{2}$$

where, p_g is the softmax output of ground-view policy logits and p_a is the softmax output of aerial-view policy logits. In this way, the learned policy could be less sensitive to differences in representation made by the convolution networks.

3.4. View Dropout

While there are two pathways and thus two sets of policy logits, the agent can sample only one action at a time. We propose to fuse the policy outputs of the two modalities through a dropout gating layer, that we call *view dropout* since it chooses over modalities instead of over individual perceptual units. This dropout layer aims at enforcing the cross-modal transferability of the agent.

3.5. Total Loss Function

The final objective is

$$\ell_{\text{total}} = \ell_{\text{RL}} + \lambda\ell_{\text{embed}} + \gamma\ell_{\text{policy}} \tag{3}$$

where ℓ_{RL} is the reinforcement learning loss. λ and γ are coefficients indicating the importance of embedding and distillation loss terms respectively. They can be set according to some prior or domain knowledge, or be the subject of hyper-parameter search.

3.6. Transfer Learning with Cross-View Policy

We present in this section that a cross-view policy can be used for transfer learning. Figure 3 illustrates the three stages of the transfer learning setting: training, adaptation and transfer. The details of each stage are explained below.

- *Training*: The agent is initially trained on n regions using paired aerial and ground view observations with L_2 loss, KL loss and view dropout. All modules (two parallel pathways of CNN, local RNNs and the policy RNN) are trained in this stage.
- *Adaptation*: At the adaptation stage, only the aerial images in the target region are used and only the *locale* LSTM (red box) is trained on the aerial-view environment. Since the ground-level view and the aerial view pathways have been already trained to share similar representations and policy actions, this stage makes the agent ready for substituting the aerial view for the ground-level view during for next phase.
- *Transfer*: During transfer, the convolution networks and *policy* LSTM of the agent are frozen, with only the target *locale* LSTM being retrained, solely on ground-view observations. The reason why the CNN and *policy* LSTM are frozen is because this modular approach efficiently avoids catastrophic forgetting in already trained city areas (as their corresponding modules are left untouched).

4. Experiments

In this section, we present our experiments and results, study the effect of curriculum and heading information, perform an ablation study for two components of the loss function, and demonstrate the need for the adaptation stage.

4.1. Setup

Goal-Driven Navigation (*Courier* Task). Following [29], the agent's task consists in reaching, as fast as possible, a goal destination specified as lat/long coordinates, by traversing a Street View graph of panoramic images that cover areas between 2km and 5km a side. Panoramas are spaced by about 10m; the agent is allowed 5 actions: move forward (only if the agent is facing another panorama, otherwise that action is wasted), turn left/right by 22.5 degrees and turn left/right by 67.5 degrees. Upon reaching the goal (within 100m tolerance), the agent receives a reward proportional to the bird flight distance from the starting position to the goal; early rewards are given if the agent is within 200m of the goal. Episodes last for 1000 steps and each time a goal is reached, a new goal location is sampled, encouraging the agent to reach the goals quickly.

Multimodal Egocentric Dataset. We build a multiview environment by extending StreetLearn [29]. Aerial images are downloaded that cover both New York City and Pittsburgh. At each lat/long coordinate, the environment returns an 84 × 84 aerial image centered at the location, of same size as the ground view image, and rotated according to the agent's heading towards North. Aerial images cover roughly 0.001 degree spatial differences in latitude and longitude. The training set is composed of four regions: Downtown NYC, Midtown NYC, Allegheny district in Pittsburgh and CMU campus nearby in Pittsburgh, while the testing region is a held-out set and located around the NYU campus and Union Square in NYC, which does not overlap with training areas (see Figure 1 for their approximate locations).

Transfer Learning Setup. The real transfer task includes three stages, *i.e.*, training, adaptation and transfer. The agent is trained in one area using both ground-view and aerial-view observations during the training stage with 1 billion steps. In the adaptation stage, the agent only takes in the aerial-view observations and retrains the local LSTM in the target transfer area with 500 million steps. Then the agent navigates in the transfer area with only ground-view observations and is continuously trained. Note that without additional aerial-view observations, an agent cannot be transferred in such a 3-stage setup.

We also conduct ablation studies by skipping the adaptation phase (see Section 4.5). In that case, the agent is trained on both views in the training regions and learns to navigates in the target region using only ground-view observations. During the transfer stage, the agent is fine-tuned in the target region.

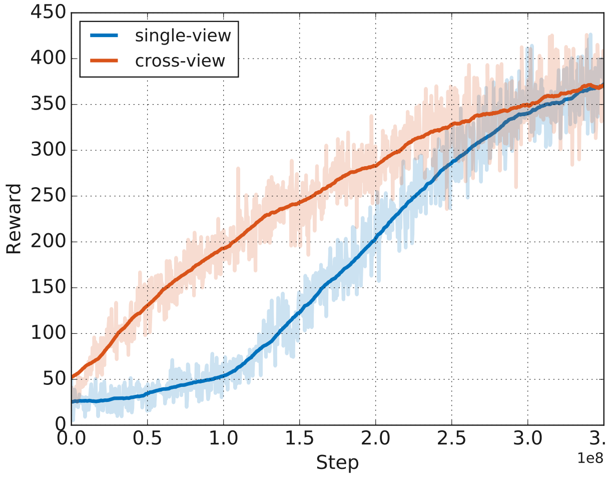


Figure 4. Rewards gained by the agent at the transfer stage in a fixed target city region. The agent is continuously trained during transfer. Higher rewards are better. The proposed cross-view approach significantly outperforms single-view baseline in terms of reward and convergence speed.

Architecture. Our model is an extension of the model used in [29] which considered only the ground-view modality. To gain intuition from results effectively, we use the same type of architectures for all networks as in [29], and compare our cross-view learning approach with the multi-city navigation agent proposed in [29] (the latter architecture corresponds to the ground-view pathway in our architecture on Figure 2).

Parameter Selection. As in [29], the batch size is 512, RMSprop is used with an initial learning rate of 0.001 and with linear decay; the coefficient of embedding and policy distillation losses were set to $\lambda = 1$ and $\gamma = 1$.

4.2. Cross-View vs. Single-View

We start by presenting the rewards in transfer stage gained by the proposed cross-view method and the baseline single-view method in Figure 4. The cross-view agent leveraged the aerial images in the adaptation stage to adapt better to the new environment, however, in transfer stage, both agents only observe the ground-view. This aligns with real world scenarios well as the top-down aerial-view is not always available in an online manner. The *locale* LSTM of the agents are being retrained during the transfer stage; all other components such as CNN and *policy* LSTM are frozen. The target region is fixed and goals are randomly sampled from this region. No curriculum is used during retraining. Heading information is not used as well since a "compass" is not always guaranteed in navigation.

Figure 4 shows the rewards obtained by cross-view and single-view methods in the transfer phase. The cross-view method achieves around 190 reward at 100M steps and

280 reward at 200M steps, both of which are significantly higher than the single-view method (50 @ 100M and 200 @ 200M). We can see on the figure that the cross-view approach significantly outperforms the single-view method in terms of learning speed. It is also worth noting that both methods eventually achieve similar rewards after 350M steps since their architectures are identical during the transfer phase and their performances are getting saturated when a large number of samples are seen.

Besides retraining, we conduct an experiment to evaluate the *zero-shot reward* or *jumpstart reward* [40], which is obtained by testing the agent in the target region without any additional retraining. The zero-shot reward is averaged over 350M steps. The proposed cross-view method achieves a zero-shot reward of 29, significantly higher than the reward of 5 obtained by the single-view method. We attribute this to the adaptation phase using the aerial-view imagery. The *total reward* is defined as the accumulated rewards from beginning to 350M steps, which measures the overall performance of the agent. Our approach achieves 87.64B, around 40% improvement over the single-view baseline (62.2B total reward). All above metrics show that our method is faster in terms of learning progress.

The above results also suggest that the proposed transfer learning allows the agent to gain knowledge about the target city region so that the subsequent navigation can start from a good initial status and such knowledge can significantly improve the continual learning of the agents.

4.3. Curriculum and Heading

As we mentioned earlier, both the training and adaptation stages utilize a pre-defined curriculum and environment-provided heading information, following [29]. The curriculum increases the distance to goals over time; so that the agent always starts from easier tasks (closer to the goals). This time, we incorporate extra heading information during training, by adding an auxiliary supervised task that consists in predicting the heading from observations. Previous transfer experiments did not utilize them because heading may not be available in a real world scenario; in this section, we examine how the curriculum and heading information could affect the performance of the agents.

Figure 5 compares transfer phase rewards for four different methods: single/cross views with curriculum, and single/cross views with both curriculum and heading prediction auxiliary tasks. The results suggest that with the heading auxiliary task, the agents can achieve significantly higher performance (approximately 450 reward at step 350M). In addition, the gap between single-view and cross-view is smaller with heading information.

We also observed that cross-view methods manage to learn irrespective of the curriculum design. In other words, our cross-view architecture compensates for the lack of curriculum by transferring knowledge between cities. Rewards in the cross-view approach grow linearly and reach around 290 at 200M steps, which is comparable with the results shown in Figure 4. However, the performance of single-view agents degrades significantly without training curriculum. It fails to reach over 50 reward within 100M steps (dark blue curve in Figure 4), 30 less than the one trained with curriculum (dark blue curve in Figure 5). Without curriculum learning, the single-view agent learns slowly.

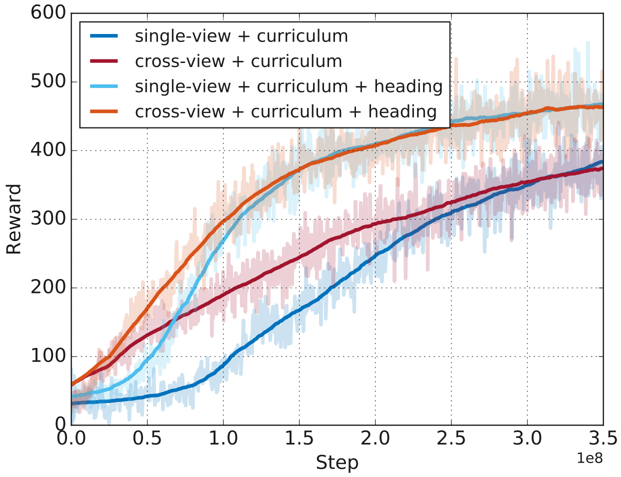


Figure 5. Transfer rewards of agents with curriculum and heading prediction auxiliary task. The performance gap between single-view and cross-view methods are smaller when heading information is used. Heading prediction also leads to higher rewards.

4.4. Adaptation Using Aerial Views

An important question is how much improvement is brought by aerial-view based transfer learning. Figure 6 compares transfer phase rewards between 1) cross-view agents that are transferred with aerial-view and 2) agents that skipped the adaptation stage. All transfers are under done using the curriculum. We also compare agents with and without heading prediction.

Figure 6 suggests that the adaptation stage is important and leads to a higher zero-shot reward, faster learning progress in the initial phase and better overall performance of the agent. The effect of adaptation becomes more significant when heading information is dropped during the adaptation stage (which fits better to real world situations). Unsurprisingly, as the agents are fully retrained, their performances become comparable after a sufficiently large number of training steps.

4.5. Ablation Study

The proposed cross-view policy learning is composed of multiple components: L_2 embedding similarity loss, KL policy distillation loss and view dropout. In this section, we

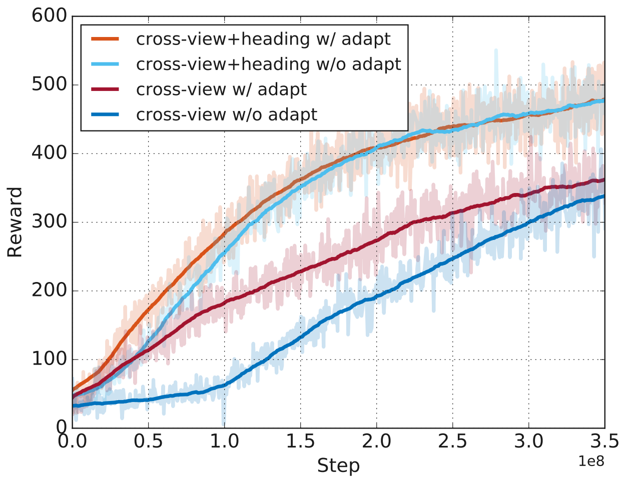


Figure 6. Rewards at the transfer phase (with curriculum learning) for cross-view agents going or not through the adaptation stage.

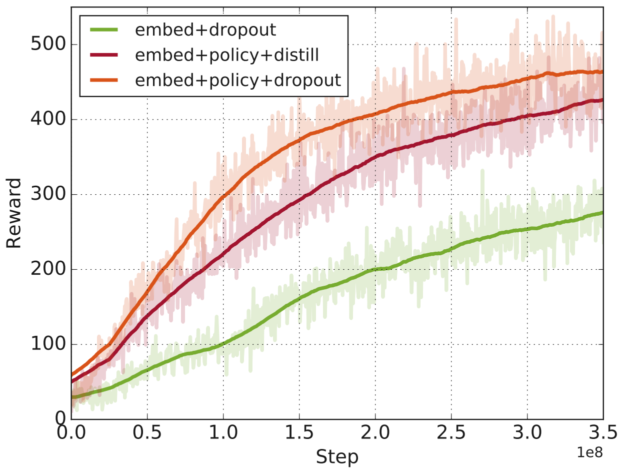


Figure 7. Ablation Study: Transfer the agents under curriculum with heading prediction auxiliary task.

evaluate the contribution of each one of those components.

In order to show the strength of view dropout, we implement another approach which uses the same L_2 distance loss between embeddings and KL divergence loss between policy logits but always taking the street-view policy logits for action selection (instead of randomly dropping either of the views). In this case, the aerial-view policy logits are not involved in decision making. We name this method "view distillation" (in short, *distill*) as an additional baseline since it reflects the setting of model distillation – one model is optimized for the main objective while the other one is optimized only to match the logits of the former.

Figure 7 shows the rewards for transfer with curriculum and heading auxiliary loss[1]. Three cross-view methods are compared: (a) full model without KL loss, (b) full model with view dropout replaced with view distillation, and (c) the full model.

According to the figure, simply using L_2 embedding loss without KL policy loss is insufficient to learn a good transferrable representation across views. Its result is significantly worse than the full model. This is probably because the discrepancy between the two views makes it impossible to project them into the same space. There are always differences in their representations and such differences are enlarged after passing through the policy networks. Having an additional KL policy loss would allow the learned policy to be more robust (or less sensitive) to such small differences in feature representations.

One may also notice that the agent (*distill*) that always uses the street-view policy for action selection could achieve decent performance but still is non-trivially worse than the agent that uses view dropout. Such results suggest that the L_2 embedding loss and the KL policy loss are able to distill a street-view agent into a good aerial-view agent. However, that distilled policy is not interchangeable across views. Training an agent with view dropout can be seen as replacing the navigation task by a more difficult task where the agent has to learn to quickly switch context at every single step. An agent trained on this harder task generalizes across observation modalities.

5. Conclusion

We proposed a generic framework for transfer learning using an auxiliary modality (or view), composed of three stages: (a) training with both modalities, (b) adaptation using an auxiliary modality and (c) transfer using the major modality. We proposed to learn a cross-view policy including learning a joint embedding space, distilling the policy across views and dropping out modalities, in order to learn representations and policies that are inter-changeable across views. We evaluated our approach on a realistic navigation environment, *StreetLearn*, and demonstrated its effectiveness by transferring navigation policies to unseen city regions.

Another extension would consist in providing the agent with the start position in addition to the goal position, so that the problem simplifies to learning to find the optimal path from A to B, without the need for learning to relocalize and to find A. After all, as it happened during the successful journey through unknown seas made by the crew of the *Endurance*, the navigator often knows their starting position, and the interesting question is how to reach the destination.

Acknowledgements. We thank Carl Doersch, Raia Hadsell, Karl Moritz Hermann, Mateusz Malinowski, and Andrew Zisserman for their discussions and contributions.

[1] The trend for transfer without heading information is very similar.

References

[1] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. D. Reid, S. Gould, and A. van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *CVPR*, pages 3674–3683. IEEE Computer Society, 2018. 3

[2] J. Ba and R. Caruana. Do deep nets really need to be deep? In *Advances in neural information processing systems*, pages 2654–2662, 2014. 3

[3] A. Banino, C. Barry, B. Uria, C. Blundell, T. Lillicrap, P. Mirowski, A. Pritzel, M. J. Chadwick, T. Degris, J. Modayil, G. Wayne, H. Soyer, F. Viola, B. Zhang, R. Goroshin, N. Rabinowitz, R. Pascanu, C. Beattie, S. Petersen, A. Sadik, S. Gaffney, H. King, K. Kavukcuoglu, D. Hassabis, R. Hadsell, and D. Kumaran. Vector-based navigation using grid-like representations in artificial agents. *Nature*, 557(7705):429, 2018. 1

[4] M. Bansal, H. S. Sawhney, H. Cheng, and K. Daniilidis. Geo-localization of street views with aerial image databases. In *Proceedings of the 19th ACM International Conference on Multimedia*, MM '11, pages 1125–1128, New York, NY, USA, 2011. ACM. 3

[5] J. Bruce, N. Sünderhauf, P. Mirowski, R. Hadsell, and M. Milford. One-shot reinforcement learning for robot navigation with interactive replay. *arXiv preprint arXiv:1711.10137*, 2017. 3

[6] D. S. Chaplot, G. Lample, K. M. Sathyendra, and R. Salakhutdinov. Transfer deep reinforcement learning in 3d environments: An empirical study. In *NIPS Deep Reinforcement Learning Workshop*, 2016. 2, 3

[7] D. S. Chaplot, K. M. Sathyendra, R. K. Pasumarthi, D. Rajagopal, and R. Salakhutdinov. Gated-attention architectures for task-oriented language grounding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017. 3

[8] H. Chen, A. Shur, D. Misra, N. Snavely, and Y. Artzi. Touchdown: Natural language navigation and spatial reasoning in visual street environments. *arXiv preprint arXiv:1811.12354*, 2018. 2

[9] C. J. Cueva and X.-X. Wei. Emergence of grid-like representations by training recurrent neural networks to perform spatial localization. *International Conference on Learning Representations*, 2018. 1

[10] A. D. Ekstrom, H. J. Spiers, V. D. Bohbot, and R. S. Rosenbaum. *Human Spatial Navigation*. Princeton University Press, 2018. 1

[11] L. Espeholt, H. Soyer, R. Munos, K. Simonyan, V. Mnih, T. Ward, Y. Doron, V. Firoiu, T. Harley, I. Dunning, et al. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2018. 4

[12] N. C. Garcia, P. Morerio, and V. Murino. Modality distillation with multiple stream networks for action recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 103–118, 2018. 3

[13] S. Gupta, D. Fouhey, S. Levine, and J. Malik. Unifying map and landmark based representations for visual navigation. *arXiv preprint arXiv:1712.08125*, 2017. 3

[14] S. Gupta, J. Hoffman, and J. Malik. Cross modal distillation for supervision transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2827–2836, 2016. 3

[15] K. M. Hermann, M. Malinowski, P. Mirowski, A. Banki-Horvath, K. Anderson, and R. Hadsell. Learning to follow directions in street view. *arXiv preprint arXiv:1903.00401*, 2019. 2, 3

[16] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 3

[17] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. 2

[18] J. Hoffman, S. Gupta, and T. Darrell. Learning with side information through modality hallucination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 826–834, 2016. 3

[19] K. Kansky, T. Silver, D. A. Mély, M. Eldawy, M. Lázaro-Gredilla, X. Lou, N. Dorfman, S. Sidor, S. Phoenix, and D. George. Schema networks: Zero-shot transfer with a generative causal model of intuitive physics. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1809–1818. JMLR. org, 2017. 3

[20] H. J. Kim, E. Dunn, and J.-M. Frahm. Learned contextual feature reweighting for image geo-localization. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3251–3260, 2017. 3

[21] E. Kolve, R. Mottaghi, D. Gordon, Y. Zhu, A. Gupta, and A. Farhadi. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*, 2017. 3

[22] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 2, 4

[23] A. Li, V. I. Morariu, and L. S. Davis. Planar structure matching under projective uncertainty for geolocation. In *European Conference on Computer Vision (ECCV)*, 2014. 3

[24] Y. Li, J. Yang, Y. Song, L. Cao, J. Luo, and L.-J. Li. Learning from noisy labels with distillation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1910–1918, 2017. 3

[25] T.-Y. Lin, Y. Cui, S. Belongie, and J. Hays. Learning deep representations for ground-to-aerial geolocalization. Boston, MA, 2015. Computer Vision and Pattern Recognition (CVPR). Oral. 3

[26] Z. Luo, J.-T. Hsieh, L. Jiang, J. Carlos Niebles, and L. Fei-Fei. Graph distillation for action detection with privileged modalities. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 166–183, 2018. 3

[27] P. Mirowski, A. Banki-Horvath, K. Anderson, D. Teplyashin, K. M. Hermann, M. Malinowski, M. K. Grimes, K. Simonyan, K. Kavukcuoglu, A. Zisserman, et al. The streetlearn environment and dataset. *arXiv preprint arXiv:1903.01292*, 2019. 1, 2, 3

[28] P. Mirowski, R. Pascanu, F. Viola, H. Soyer, A. J. Ballard, A. Banino, M. Denil, R. Goroshin, L. Sifre, K. Kavukcuoglu, et al. Learning to navigate in complex environments. In *International Conference on Learning Representations (ICLR)*, 2017. 3

[29] P. W. Mirowski, M. K. Grimes, M. Malinowski, K. M. Hermann, K. Anderson, D. Teplyashin, K. Simonyan, K. Kavukcuoglu, A. Zisserman, and R. Hadsell. Learning to navigate in cities without a map. In *NeurIPS*, 2018. 2, 3, 4, 6, 7

[30] S.-I. Mirzadeh, M. Farajtabar, A. Li, and H. Ghasemzadeh. Improved knowledge distillation via teacher assistant: Bridging the gap between student and teacher. *arXiv preprint arXiv:1902.03393*, 2019. 3

[31] K. Mo, H. Li, Z. Lin, and J.-Y. Lee. The adobeindoornav dataset: Towards deep reinforcement learning based real-world indoor robot visual navigation. *arXiv preprint arXiv:1802.08824*, 2018. 1, 2, 3

[32] A. Mousavian, A. Toshev, M. Fiser, J. Kosecka, and J. Davidson. Visual representations for semantic target driven navigation. *CoRR*, abs/1805.06066, 2018. 1

[33] J. Oh, V. Chockalingam, S. Singh, and H. Lee. Control of memory, active perception, and action in Minecraft. *International Conference on Machine Learning (ICML)*, 2016. 3

[34] Ö. C. Özcanli, Y. Dong, and J. L. Mundy. Geo-localization using volumetric representations of overhead imagery. *International Journal of Computer Vision*, 116:226–246, 2015. 3

[35] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010. 3

[36] E. Parisotto, D. S. Chaplot, J. Zhang, and R. Salakhutdinov. Global pose estimation with an attention-based recurrent network. *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops*, 2018. 3

[37] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014. 3

[38] S. Shah, D. Dey, C. Lovett, and A. Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and service robotics*, pages 621–635. Springer, 2018. 1, 3

[39] L. Tai, G. Paolo, and M. Liu. Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 31–36. IEEE, 2017. 1, 2, 3

[40] M. E. Taylor and P. Stone. Transfer learning for reinforcement learning domains: A survey. *J. Mach. Learn. Res.*, 10:1633–1685, Dec. 2009. 7

[41] Y. Tian, C. Chen, and M. Shah. Cross-view image matching for geo-localization in urban environments. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1998–2006, 2017. 3

[42] X. Wang, Q. Huang, A. Celikyilmaz, J. Gao, D. Shen, Y.-F. Wang, W. Y. Wang, and L. Zhang. Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. *arXiv preprint arXiv:1811.10092*, 2018. 3

[43] G. Wayne, C. Hung, D. Amos, M. Mirza, A. Ahuja, A. Grabska-Barwinska, J. W. Rae, P. Mirowski, J. Z. Leibo, A. Santoro, M. Gemici, M. Reynolds, T. Harley, J. Abramson, S. Mohamed, D. J. Rezende, D. Saxton, A. Cain, C. Hillier, D. Silver, K. Kavukcuoglu, M. Botvinick, D. Hassabis, and T. P. Lillicrap. Unsupervised predictive memory in a goal-directed agent. *arXiv preprint arXiv:1803.10760*, 2018. 2, 3

[44] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992. 4

[45] R. W. Wolcott and R. M. Eustice. Visual localization within lidar maps for automated urban driving. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 176–183. IEEE, 2014. 1

[46] J. Zhang, J. T. Springenberg, J. Boedecker, and W. Burgard. Deep reinforcement learning with successor features for navigation across similar environments. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2371–2378. IEEE, 2017. 3

[47] J. Zhang, L. Tai, J. Boedecker, W. Burgard, and M. Liu. Neural slam: Learning to explore with external memory. *arXiv preprint arXiv:1706.09520*, 2017. 2, 3

[48] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu. Deep mutual learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4320–4328, 2018. 3

[49] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 3357–3364. IEEE, 2017. 1, 2, 3