
Ziad Al-Halah\textsuperscript{1} \quad Santhosh K. Ramakrishnan\textsuperscript{1,2} \quad Kristen Grauman\textsuperscript{1,2}
\textsuperscript{1}The University of Texas at Austin \quad \textsuperscript{2}Meta AI
ziadlhhl@gmail.com, srama@cs.utexas.edu, grauman@cs.utexas.edu

Abstract

In reinforcement learning for visual navigation, it is common to develop a model for each new task, and train that model from scratch with task-specific interactions in 3D environments. However, this process is expensive; massive amounts of interactions are needed for the model to generalize well. Moreover, this process is repeated whenever there is a change in the task type or the goal modality. We present a unified approach to visual navigation using a novel modular transfer learning model. Our model can effectively leverage its experience from one source task and apply it to multiple target tasks (e.g., ObjectNav, RoomNav, ViewNav) with various goal modalities (e.g., image, sketch, audio, label). Furthermore, our model enables zero-shot experience learning, whereby it can solve the target tasks without receiving any task-specific interactive training. Our experiments on multiple photorealistic datasets and challenging tasks show that our approach learns faster, generalizes better, and outperforms SoTA models by a significant margin. Project page: https://vision.cs.utexas.edu/projects/zsel/

1. Introduction

In visual navigation, an agent must intelligently move around in an unfamiliar environment to reach a goal, using its egocentric camera to avoid obstacles and decide where to go next. As a fundamental research problem in embodied AI, visual navigation has many potential applications—such as service robots in the home or workplace, mobile search and rescue robots, assistive technology for the visually impaired, and augmented reality systems to help people navigate or find objects.

Recent work in computer vision explores visual navigation from many different fronts. In PointNav, an agent is asked to go to a specific position in an unmapped environment (e.g., go to \((x, y)\)) \cite{6,56,57}. In ObjectNav, the agent must find an object by name (e.g., go to the nearest telephone) \cite{6,9}. In RoomNav, the agent must find a room (e.g., go to the kitchen) \cite{49,56,67}. In AudioNav, the agent must find a sounding target (e.g., find the ringing phone) \cite{16,28}. In ImageNav, the agent must go to where a given photo was taken \cite{14,55,75}. Each case presents a distinct goal to the agent. Accordingly, researchers have pursued task-specific models to treat each one, typically training policies with deep reinforcement learning (RL).

Despite exciting advances, learning task-specific navigation policies has inherent limitations. Training embodied agents from scratch for each new task and relying on special-purpose architectures and priors (e.g., room layout maps for RoomNav, object co-occurrence priors for ObjectNav, directional cues for AudioNav, etc.) requires repeated access to training environments for gathering new agent ex-
experience in the context of each task, greatly hindering sample efficiency. Even with today’s fast simulators and photorealistic scanned environments [12, 57, 68], this typically amounts to days and weeks of computation on a small army of GPU servers to train a single policy. Moreover, by tackling each variant in isolation, agents fail to capture what is common across the tasks. Finally, some tasks require manual annotations such as object labels in 3D space, which naturally limits how extensively they can be trained.

In this work, we challenge the assumption that distinct navigation tasks require distinct policies. Intuitively, finding a good policy for one navigation task should help with the rest. For example, if we know how to find a microwave, then finding a kitchen should be easy too; if we know how to find an object by name, then finding it based on a hand-drawn sketch—or the sounds it emits—should be possible too. In short, it should be beneficial to learn one navigation task and then apply the accumulated experience to many.

To that end, we propose a modular transfer learning approach for semantic visual navigation that enables zero-shot experience learning. See Figure 1. First, we develop a general-purpose semantic search policy. Specifically, using a novel reward and task augmentation strategy, we train a source policy for the image-goal task, where the agent receives a picture taken at some unknown camera pose somewhere in the environment, and must travel to find it. Next, we develop a joint goal embedding that is trained offline (i.e., no interactive agent experience) to relate various target goal types to image-goals. Finally, we address target downstream tasks either by zero-shot transfer with no new agent experience, or by fine-tuning with a limited amount of agent experience on the target task.

Zero-shot learning traditionally focuses on supervised tasks such as image recognition [5, 38, 69], where models forgo using labeled samples for the new class. Instead, the proposed zero-shot experience learning (ZSEL) focuses on reinforcement learning tasks, where models forgo using interactions in the physical environment for the new navigation task. ZSEL is important for lifelong learning, where an agent will face novel tasks once it is deployed and must solve them while using no or few training episodes.

Using hundreds of multi-room environments from Matterport3D [12], Gibson [68], and HM3D [52], we demonstrate our approach for four challenging tasks and goals expressed with five different modalities—images, category names, audio, hand-drawn sketches, and egomaps. Our ImageNav results advance the state of the art, and our modular transfer approach outperforms the best existing methods of transfer based on self-supervision, supervision, and RL. Finally, our ZSEL performance on 5 semantic navigation tasks is equivalent to 507 million interactions required by task-specific policies learned from scratch.

2. Related Work

Visual Navigation Traditional methods in visual navigation often rely on mapping the 3D space and then planning their movements [8, 27, 63]. However, fueled by fast simulators [35, 57] and large-scale photorealistic datasets [12, 16, 52, 68] there have been great advances in learning-based navigation approaches [14, 17, 50] leading to a near-perfect agent for tasks like point-goal navigation [65]. In this work, we consider semantic visual navigation, where the agent is given a semantic description of the goal (e.g., object-goal [6, 9, 51], image-goal [14, 75], room-goal [67], audio-goal [15, 16]) but, unlike point-goal, the goal location is unknown. Hence, the agent needs to leverage learned scene priors to explore the environment efficiently to find and navigate to the target. Current approaches tackle each navigation task separately: a new model is trained for each task and each target modality [9, 14, 15, 67], which has the disadvantages discussed above. In contrast, we propose a unified approach to semantic visual navigation, in which a single trained policy can handle diverse tasks and goal modalities.

Transfer Learning in Navigation Pre-learning a representation from large-scale image datasets [21, 43] and transferring it to a downstream task proved to be very successful for visual recognition [10, 18, 19, 29, 73]. We observe a similar trend in embodied navigation, where pre-learning good representations of the 3D environment [47, 48, 53, 70, 74] or primitive skills [25, 30, 42, 65] help the agent to learn a downstream task better while using fewer training samples. Recent methods focus on pre-training the observation encoder of the agent, either in a supervised [30, 39, 58, 62, 72] or self-supervised [19, 22] fashion. While this leads to improved performance on the target tasks, a new policy is still learned from scratch for each task, resulting in low sample efficiency. In contrast, our approach enables a full transfer paradigm where all the agent’s components can be reused efficiently on the downstream tasks. Prior work shows that transferring a strong point-goal policy to non-goal driven tasks (e.g., flee and exploration) can lead to better performance [65]. Differently, we propose to learn and transfer a general-purpose semantic search policy. Our policy can find semantic goals presented in different modalities for a diverse set of goal-driven navigation tasks.

Sharing knowledge between multiple tasks can be achieved in a multi-task learning setup [13, 64] where all tasks are learned jointly in a supervised manner, or via meta-RL [26, 66] where a meta policy learned from a distribution of tasks is finetuned on the target. Unlike these methods, our policy is learned from one task that does not require manual annotations, and it can be transferred in a zero-shot setup where the policy does not receive any interactive training on the target.

Zero-Shot Learning Zero-shot learning (ZSL) can be seen as an extreme case of transfer learning where the
target task has zero training samples. Prior ZSL work focuses on supervised learning, e.g., image classification [4, 5, 24, 38, 40, 54, 69]. In contrast, the proposed zero-shot experience learning (ZSEL) setup learns behaviors rather than classifiers; the policy learned on a source task needs to perform a set of target tasks, without receiving any new interactive experiences on the target. Further, unlike [60] where a world model for synthetic environments is constructed, and control policies are trained on ‘imagined’ episodes, we consider a model-free approach and a ZSEL setup in realistic environments where the policy receives zero target interactions (i.e., neither imagined nor real). To our knowledge, we are first to propose a ZSEL model for embodied navigation.

3. Plug & Play Modular Transfer Learning

We introduce a novel transfer learning approach for visual navigation. Our model has three main components: 1) we start by learning a semantic search policy for image goals using a novel reward and task augmentation (Fig. 2a); 2) we leverage the image goal encoder to learn a joint goal embedding space for the different goal modalities (Fig. 2b); and finally, 3) we transfer the learned agent modules to downstream tasks in a plug and play fashion (Fig. 2c).

In the following, we consider an agent with 3 main modules: 1) an observation encoder ($f_O$) that encodes the received observations $o_t$ from the environment; 2) a goal encoder ($f_G$) that encodes the task’s goal; 3) a policy ($\pi$) that uses the output of $f_O$ and $f_G$ to navigate and find the goal.

3.1. Semantic Search Policy for Image Goals

The policy is a key component in the modern end-to-end visual navigation agent. It guides the agent towards solving a task given a set of sequential observations and a goal. Such policies are often learned with reinforcement learning (RL) where the agent interacts with its environment (by moving about) and attempts to solve the task in a trial and error fashion. If the agent succeeds in its attempt, then it receives a reward to encourage such behavior from the policy in the future.

A main challenge for this learning paradigm is that the policy requires a large number of interactions with the environment in order to find a proper way to solve the task. This usually amounts to tens and hundreds of millions [44, 45] and up to billions [65] of interactions, and correspondingly days or weeks of GPU cluster time. Furthermore, for each new task a policy is typically learned from scratch, which further increases the learning cost substantially.

We propose to learn a general-purpose semantic search policy that can be transferred and perform well on a variety of navigation tasks. Our idea is to learn such a policy with the image-goal task, where the agent receives a picture taken at some unknown camera pose somewhere in the environment, and must travel to find it. Our choice of image-goal for the source policy is significant. It requires no manual annotations, and image-goals can be sampled freely anywhere in a training environment. As a result, the policy can be trained on large-scale experience (e.g., collected from a fleet of robots deployed in various environments) which can improve its generalization to new tasks and domains. Furthermore, an image-goal encourages the learned policy to capture semantic priors for finding things in a 3D space. For example, by seeking images of couches and chairs, the agent learns implicitly to leverage these objects’ context and the room layout in order to find the image views effectively.

Task Definition

In an episode of image goal navigation, the agent starts from a random position $p_0$ in an unexplored scene, and it is tasked to find a certain location $p_G$ given an image $I_G$ sampled with the camera at $p_G$. The agent receives an RGB observation $o_t$ at each step $t$ and needs to perform the best sequence of actions $a_t \in \{\text{move forward, turn left, turn right, stop}\}$ that would bring it to the goal within a maximum number of steps $S$. Unlike the common point-goal task where the goal location is known [6], here $p_G$ is unknown, and the agent needs to leverage the learned semantic priors to search and find where $I_G$ could have been sampled from.

Our setup differs from recent methods in ImageNav where panoramic 360° FoV sensors are required [14, 37, 45]. Here, we consider a standard 90° FoV for the agent’s view [75]. While having a complete FoV sensor simplifies localization, this strong requirement is often not available in common robotic platforms [1–3] and leads to high computational cost. This reduces the scalability and adoption of such methods by diverse agent configurations. In addition, our task setup allows our model to transfer to a diverse set of semantic navigation tasks in a plug and play fashion without the need for modifications to the target tasks (for which the literature does not use panoramic images).

View Reward

It is common to use the reduced distance to the goal to reward the agent for getting closer to $p_G$ in addition to the success reward of finding and stopping within a small distance $d_s$ of $p_G$. However, while this reward proves to be quite successful for navigation tasks like point-goal, we argue it is less suited for semantic goals like images. Since the reward does not carry a signal about the semantic goal itself, the agent may fail or require much more experience in order to capture the implicit relation between the goal and the distance to goal reward (DTG). For example, if the goal shows an image of an oven, the agent may get close and stop nearby while looking at a book on the counter, and nonetheless receive a full success reward. This may lead to capturing trivial or incoherent associations between the goal and the agent’s observations.
In order to encourage the agent to leverage the information captured in the goal description $I_G$ and effectively capture useful semantic priors that may help it in finding $p_G$, we propose a new reward function that rewards the agent for looking at $I_G$ when getting closer to $p_G$, so it can better draw the association between its $o_t$ and $I_G$. Specifically, we define the reward function at step $t$ as:

$$ r_t = r_d(d_t, d_{t-1}) + [d_t \leq d_s] \cdot r_\alpha(\alpha_t, \alpha_{t-1}) - \gamma $$

(1)

where $r_d$ is the reduced distance to the goal from the current position relative to the previous one, $r_\alpha$ is the reduced angle in radians to the goal view from the current view relative to the previous one, $[\cdot]$ is the indicator function, and $\gamma = 0.01$ is a slack reward to encourage efficiency. Note, this reward will encourage the agent to look at $I_G$ when it gets near the goal, since it is rewarded to reduce the angle between its current view $v_t$ and the view of the goal $v_G$ (see Fig. 2a). Finally, the agent receives a maximum success reward of 10 if it reaches the goal and stops within a distance of $d_s$ from $p_G$ and an angle $\alpha_s$ from $v_G$:

$$ R_s = 5 \times ([d_t \leq d_s] + [d_t \leq d_s \text{ and } \alpha_t \leq \alpha_s]) $$

(2)

We set $d_s = 1$ m (the success distance of the task) and $\alpha_s = 25^\circ$ to allow for a good overlap between $v_t$ and $v_G$ and enable the agent to draw the association between its observation $o_t$ and the goal $I_G$.

**View Augmentation** In addition to the view reward introduced above, we also provide a simple task augmentation method to promote generalization by increasing the diversity of goals presented to the agent. For each training episode, rather than having a fixed $I_G$, we sample a view from a random angle at location $p_G$ and provide the agent with the associated $I_G$ from the sampled view as the goal descriptor. This has a regularization effect on the model learning; the agent will be less likely to overfit due to the changing goal description each time the agent experiences a given goal. Furthermore, with the start $p_0$ and the goal location $p_G$ fixed in a training episode but not $I_G$, this encourages the agent to capture implicit spatial semantic priors of things that usually appear near each other as viewed from $p_G$. For example, the agent would learn that an image of chairs as seen when peeking from the door is likely to be at the same location of the current image goal that is showing a dining table, since the agent experienced the same episode before but with $I_G$ showing the chairs, hence prompting the agent to explore the dining room.

**Policy Training** We train our policy using reinforcement learning (RL), Fig. 2a. For each training episode, we sample an image-goal $I_G$ from $p_G$. The agent encodes its current observation $o_t$ (an RGB image) with $f_O$ and the image-goal with $f_G^I$ and passes these encodings to the policy $\pi$. The policy further encodes these information along with the history of observations so far to produce a state embedding $s_t$. An actor-critic network leverages $s_t$ to predict state value $c_t$ and the agent’s next action $a_t$. Based on the agent’s state in the environment, it receives a reward (Eq. 1 and Eq. 2). The model is trained end-to-end using PPO [59].

### 3.2. Joint Goal Embedding Learning

Having learned the semantic search policy, we can now transfer our model to downstream tasks. Specifically, we consider downstream navigation tasks where the goals are *object categories* (ObjectNav [9]), *room types* (RoomNav [67]), or *view encodings* (ViewNav), and they may be expressed by the modalities of a label name, a sketch, an audio clip, or an edgemap; see Sec. 4.2. A key advantage of learning the semantic search policy using RGB image-goals is that these goals contain rich information about the target visual appearance and context. Furthermore, in order to solve the image-goal navigation task, our model learns to encode these visual cues via a compact dense representation produced by the image-goal encoder $f_G^I$. Our idea is to leverage $f_G^I$ to learn a joint embedding space of different goal modalities for the various tasks. In other words, we upgrade the image-goal embedding space
to be a joint goal embedding space to draw associations between the images and the different goal modalities like sketches, category names, and audio (Fig. 2b). This step can be carried out quite efficiently and using an offline dataset. For example, to learn about object-goals that are represented with a label (e.g., a chair) we only need to annotate a set of images with chairs. Then we train an object-goal encoder to produce an embedding similar to the image-goal encoder for compatible image-label pairs. In our experiments, we use offline datasets of size 20K images or less in which the “annotations” are actually automatic object detections. This is several orders of magnitude smaller than the amount of interactions usually needed to train a target-specific policy (tens to hundreds of millions) [44, 71].

Formally, let $D = \{(x_i, g_i)\}$ be a set of images $x_i$ and their associated goals $g_i$, where $g_i$ can be of any goal modality (e.g., audio, sketch, image, category name, edgemap) depending on the downstream task specifications. We learn a joint goal embedding space by minimizing the loss:

$$L(x_i, g_i) = \begin{cases} 1 - \cos(f^I_G(x_i), f^M_G(g_i)), & \text{if } y_i = +1 \\ \max(0, \cos(f^I_G(x_i), f^M_G(g_i))), & \text{if } y_i = -1 \end{cases}$$

(3)

where $f^M_G(\cdot)$ is the new goal encoder of modality $M$, $\cos(\cdot, \cdot)$ is the cosine similarity between two embeddings, and $y_i$ indicates whether the pair $(x_i, g_i)$ is similar or not, as derived from the offline annotations (e.g., chair and a picture of a chair are similar; the audio of TV and a picture of the TV are similar).

During goal embedding learning, we freeze $f^I_G$ and learn $f^M_G$ using Eq. 3 such that $f^M_G$ learns to encode its goal similar to the corresponding image embedding from $f^I_G$.

### 3.3. Transfer and Zero-Shot Experience Learning

Having learned the semantic search policy and the joint goal embedding as described above, now we can transfer our model to downstream navigation tasks (Fig. 2c). For that, we only need to replace $f^I_G$ with the suitable goal encoder for the task, such that:

$$a_t \sim \pi(f^I_O(o_t), f^M_G(g)).$$

(4)

Our plug and play modular transfer approach has multiple advantages. Since all modules are compatible with each other, this means the model can perform the target task out of the box, i.e., it does not require any further task-specific interactions to solve the target task. We refer to this setup as zero-shot experience learning (ZSEL). Training policies with modern RL frameworks is the most expensive part of the model learning, and with ZSEL we manage to circumvent this requirement. Furthermore, due to the modular nature of our approach, it is easy to generalize to a wide variety of tasks and goal modalities. For a new task, only the respective goal encoder is trained with an offline dataset then integrated in the full model in a plug&play fashion.

Finally, our model can be easily finetuned for the downstream task to capture any additional cues specific to the task to reach a better performance. Unlike the common approach in the literature where only $f_O$ is pretrained and transferred [22, 30, 58, 74], here the full model is transferred to the target task. This leads to higher initial performance, faster convergence, and better overall performance, as we will show in Sec. 4.

### 4. Evaluation

In the following experiments, we first evaluate our semantic search policy performance in the source task (image-goal navigation) compared to state-of-the-art methods (Sec. 4.1); then we show how our model transfers to a diverse set of downstream navigation tasks (Sec. 4.2).

#### Shared Implementation Setup

For fair comparisons, we adopt the same architecture and training pipeline for our model and all RL baselines, and we note any deviations from this shared setup in the respective sections. We use a ResNet9 [32, 61] for $f_O$ and a GRU [20] of 2 layers and embedding size of 128 for $f_G$. For goal encoders $f_G$, we use a ResNet9 to encode image-, sketch-, edgemap- and audio-goal modalities. We transform an audio clip to a spectrogram before encoding it by $f_G$. If the goal is a category name, we use a 2 layer MLP for $f_G$. We train the policy using DD-PPO [65] and allocate the same computation resources for all models. We use input augmentation (random cropping and color-jitter) during training to improve the stability and performance of the RL methods [36, 41, 45]. See Supp for more details. We adopt end-to-end RNN-based RL since it is a common [22, 30, 45, 58, 65, 74, 75], generic architecture, does not require hand-crafted modules, shows good performance on real-data [11, 46], and learned in sim has the potential to generalize well to real [34]. However, our contributions are orthogonal to the RL architecture used.

#### Agent Configuration

The action space of the agent consists of move forward by 25 cm, turn left and right by 30°, and stop. The agent uses only RGB observations of $128 \times 128$ resolution and $90^\circ$ FoV sensor.

#### 4.1. Image-Goal Navigation

##### Task Setup

We adopt the image-goal task as defined in Sec. 3.1. We set $S = 1000$ and $d_s = 1$ m from $p_G$.

##### Datasets

We use the Habitat simulator [57] and the Gibson [68] environments to train our model. We use the dataset from [45]. The training split contains 9K episodes sampled from each of the 72 training scenes. Following the setup from [45], all RL models are trained for 50K updates (500 million frames) on the training split. The test split has 4.2K episodes sampled uniformly from 14 disjoint (unseen) scenes. For direct comparison with [31], we also test our model on a second split (“split B”) provided by [31]
that has 3K episodes and the same structure as the test split from [45] (“split A”).

**Baselines** We compare our image-goal model to the following baselines and SoTA methods: 1) **Imitation Learning**: This model’s policy is trained using supervised learning to predict the ground truth best action on the shortest path to the goal given its current observation. 2) **Zhu et al. [75]**: The model uses a ResNet50 shared between \( f_O \) and \( f_G \), pretrained on ImageNet and frozen. 3) **Mezghani et al. [45]**: This is the SoTA panoramic image-goal navigation model. It uses a ResNet18 for \( f_O \) and \( f_G \), a 2 layer LSTM [33] for \( \pi \), and a specialized episodic memory. We adapt this model to our 90° FoV for \( o_s \) and \( I_G \) and train it using the author’s code. 4) **DTG-RL**: This model uses the shared architecture along with the common distance to goal dense reward for training. 5) **Hahn et al. [31]**: This model learns from a passive dataset of videos collected from the Gibson training scenes and uses a customized architecture based on topological maps (see [31] for details).

<table>
<thead>
<tr>
<th>Model</th>
<th>Split</th>
<th>Succ.</th>
<th>SPL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Imitation Learning</td>
<td>A</td>
<td>9.9</td>
<td>9.5</td>
</tr>
<tr>
<td>Zhu et al. [75]</td>
<td>A</td>
<td>19.6</td>
<td>14.5</td>
</tr>
<tr>
<td>Mezghani et al. [45] w/ 90° FoV</td>
<td>A</td>
<td>9.0</td>
<td>6.0</td>
</tr>
<tr>
<td>DTG-RL</td>
<td>A</td>
<td>22.6</td>
<td>18.0</td>
</tr>
<tr>
<td>Ours</td>
<td>A</td>
<td>29.2</td>
<td>21.6</td>
</tr>
<tr>
<td>Ours (View Aug. Only)</td>
<td>A</td>
<td>22.0</td>
<td>18.8</td>
</tr>
<tr>
<td>Ours (View Reward Only)</td>
<td>A</td>
<td>24.4</td>
<td>17.3</td>
</tr>
<tr>
<td>Hahn et al. [31]</td>
<td>B</td>
<td>24.0</td>
<td>12.4</td>
</tr>
<tr>
<td>Ours</td>
<td>B</td>
<td>33.0</td>
<td>23.6</td>
</tr>
<tr>
<td>Hahn et al. [31] w/ noisy actuation</td>
<td>B</td>
<td>20.3</td>
<td>8.8</td>
</tr>
<tr>
<td>Ours w/ noisy actuation</td>
<td>B</td>
<td>25.9</td>
<td>17.6</td>
</tr>
</tbody>
</table>

Table 1. Image-goal navigation results on Gibson [68].

**Results and Analysis** Table 1 reports the overall performance in terms of average success rate (Succ) and Success weighted by inverse Path Length (SPL) over 3 random seeds. Our model outperforms strong baselines and the SoTA in image-goal navigation by a significant margin. In split A, our model gains +6.6% in Succ and +2.6% in SPL over the best baseline. The method designed for panoramic sensors [45] tends to underperform in this challenging setting. We see a drop in Succ from 69% [45] to 9% when using 360° and 90° FoV, respectively, since such methods rely heavily on the 360° FoV for accurate localization. In split B, our model gains +9% in Succ and +11.2% in SPL over [31]. It is important to note that the model from [31] uses a much more complete sensor configuration than our method (pose sensor, RGB and Depth sensors of 480 × 640 resolution, and a 120° FoV) and it is trained offline from passive videos sampled from the simulator. Nonetheless, our model outperforms [31] by a large margin, showing that interactive learning of end-to-end RL models still has an advantage over heuristic and passive approaches.

**Cross-Domain Generalization** Next we test the models trained on Gibson on datasets from Matterport3D (MP3D) [12] and HM3D [52]. In addition to the visual domain gap between these datasets, MP3D has more complex and larger scenes than Gibson, and HM3D has high diversity in terms of scene types. This poses a very challenging cross-domain evaluation setting. The test split from each dataset has in total 3K episodes sampled uniformly from 100 and 18 scenes for HM3D and MP3D, respectively. Table 2 shows the results. Overall, we see a drop in performance for all models in this challenging setting, especially for HM3D since there is high diversity in the 100 test scenes. Nonetheless, our model outperforms all baselines on both datasets, showing that our contributions lead to better generalization by encouraging the agent to pay closer attention to the semantic information provided by the goal.

<table>
<thead>
<tr>
<th>Model</th>
<th>MP3D [12]</th>
<th>HM3D [52]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Succ.</td>
<td>SPL</td>
<td>Succ.</td>
</tr>
<tr>
<td>Imitation Learning</td>
<td>5.3</td>
<td>5.1</td>
</tr>
<tr>
<td>Zhu et al. [75]</td>
<td>9.8</td>
<td>7.9</td>
</tr>
<tr>
<td>Mezghani et al. [45] w/ 90° FoV</td>
<td>6.9</td>
<td>3.9</td>
</tr>
<tr>
<td>DTG-RL</td>
<td>11.0</td>
<td>9.0</td>
</tr>
<tr>
<td>Hahn et al. [31]</td>
<td>9.3</td>
<td>5.2</td>
</tr>
<tr>
<td>Ours</td>
<td>14.6</td>
<td>10.8</td>
</tr>
</tbody>
</table>

Table 2. Image-goal navigation results on MP3D and HM3D in a cross-domain evaluation setup.

**Ablations** To validate our contributions from Sec. 3.1, we test our model performance when removing the view reward or the view augmentation. As shown in Table 1, we see a degradation in performance whenever one of these components are removed, and the largest gain is realized when they work in tandem. Additionally, we test our model under noisy actuation. While methods in split A do not provide results under noisy conditions, [31] does. Following the setup from [31], we use the noise model from [14] that simulates actions learned from a Locobot [3]. Our model shows robustness to noise and maintains its advantage over the baselines (Table 1 bottom).

**4.2. Transfer to Downstream Tasks**

**Tasks** We consider 3 target tasks and 4 goal modalities: 1) **ObjectNav**: The agent is asked to find the nearest instance of one of 6 categories (bed, chair, couch, potted-plant, toilet, and tv) specified by the goal. We extend the standard ObjectNav specification [9] where the goal is given by its label (e.g., find a chair) to goals specified by a hand drawn sketches of the category, or audio produced by the object (e.g., sounds from TV). In the beginning of an episode, the agent gets either a label, a sketch, or a 4 second audio clip from a random category. An episode is successful if the agent stops within 1 m of the goal while using less than \( S = 500 \) steps. For sketches, we use images of the object categories from the Sketch dataset [23]. For audio
clips we sample sounds from the audio dataset in [15] and the audio heard by the agent is scaled by the distance to the goal (i.e., further away goals have fainter sounds). While for label goals, the category name is the same during training and testing, for audio and sketches, the goal instances used during training are disjoint from those used in testing. This poses another challenging dimension for the agent to generalize across in addition to the unseen test scenes.

2) **RoomNav:** The agent is tasked with finding the nearest room of 6 types: living-room, kitchen, bedroom, office, bathroom, and dining-room. The goal is a label (find an office) and the episode is successful if the agent steps inside the room with the maximum episode length $S = 500$ [49].

3) **ViewNav:** This task is similar to the image-goal navigation task considered above, except a different modality (an edgemap) represents the goal. This helps quantify the model performance in target tasks that are more aligned with the source task but with a substantially different goal modality. In each episode, the agent receives an edgemap from a random view in the scene and needs to find the goal and stops within 1 m to succeed. We set $S = 1000$ to give the agent enough time to succeed in this challenging task.

**Datasets** For all target tasks, we use 24 train / 5 test scenes from the Gibson [58] tiny set that has semantic annotations [7]. These scenes are disjoint from those used in Sec. 4.1. We train all methods for up to 20 million steps on the target tasks and report evaluation performance averaged over 3 random seeds. See Supp for details.

To train the goal embedding for the modalities in ObjectNav and RoomNav, we sample 14K images of objects and 20K of rooms from the training scenes. We use the object labels generated by a model from [7] to draw the associations between the images and each modality. For ViewNav, we sample 170K views from the training scenes and generate their edgemaps using an edge texture model [73]. The number of samples for the offline dataset is driven by the available instances of each goal type in the training scenes. While there is a finite number of rooms and objects, we can sample views freely from any location in a scene.

---

**Table 3. Transfer learning success rate on downstream semantic navigation tasks.**

<table>
<thead>
<tr>
<th>Model</th>
<th>Source Task</th>
<th>ObjectNav</th>
<th>RoomNav</th>
<th>ViewNav</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task Expert</td>
<td>-</td>
<td>8.0</td>
<td>6.7</td>
<td>6.6</td>
</tr>
<tr>
<td>MoCo v2 [19]</td>
<td>SSL</td>
<td>10.5</td>
<td>9.9</td>
<td>8.8</td>
</tr>
<tr>
<td>MoCo v2 [19]</td>
<td>SSL</td>
<td>7.8</td>
<td>12.7</td>
<td>11.5</td>
</tr>
<tr>
<td>Visual Priors</td>
<td>SL</td>
<td>9.3</td>
<td>9.9</td>
<td>9.1</td>
</tr>
<tr>
<td>Zhou et al. [74]</td>
<td>SL</td>
<td>15.6</td>
<td>7.6</td>
<td>9.6</td>
</tr>
<tr>
<td>CRL [22]</td>
<td>RL</td>
<td>1.9</td>
<td>0.5</td>
<td>1.0</td>
</tr>
<tr>
<td>SplitNet [30]</td>
<td>RL</td>
<td>9.0</td>
<td>6.5</td>
<td>8.8</td>
</tr>
<tr>
<td>DD-PPO (PN)</td>
<td>RL</td>
<td>13.9</td>
<td>13.6</td>
<td>12.9</td>
</tr>
<tr>
<td>Ours (ZSEL)</td>
<td>RL</td>
<td>11.3</td>
<td>11.4</td>
<td>4.4</td>
</tr>
<tr>
<td>Ours</td>
<td>RL</td>
<td>21.9</td>
<td>22.0</td>
<td>18.0</td>
</tr>
</tbody>
</table>

Figure 3. Transfer learning and ZSEL performance on downstream navigation tasks. See Supp for all tasks and modalities.

**Baselines** We compare our model to a set of baselines and SoTA models in transfer learning: 1) **Task Expert** which learns from scratch on the downstream task. 2) **MoCo v2** [19] initializes $f_O$ using MoCo training on ImageNet (IMN) or on a set of images randomly sampled from the Gibson (Gib.) training scenes. 3) **CRL** [22] pretrains $f_O$ (ResNet50) using a combination of curiosity-based exploration and self-supervised learning. We initialize $f_O$ from a pretrained model provided by the authors. 4) **Visual Priors** [58] uses a set of 4 ResNet50s pretrained encoders as $f_O$. The encoders are trained in a supervised manner to predict 4 features (e.g., semantic segmentation, surface normal) that provide maximum coverage for downstream navigation tasks [58]. 5) **Zhou et al.** [74] transfers 2 pretrained ResNet50s for depth prediction and semantic segmentation; however, unlike [58], these are used with an RGB encoder (ResNet9) trained from scratch. 6) **SplitNet** [30] pretrains $f_O$ (customized CNN) using a mix of 6 auxiliary tasks (motion and visual tasks) and point-goal navigation. We initialize $f_O$ from a pretrained model provided by the authors. 7) **DD-PPO (PN)** [65] pretrains the model for point-goal navigation (PN) and both $f_O$ and $\pi$ are transferred.

**Transfer Learning** Table 3 shows the results. Our approach outperforms all baselines by a significant margin. Interestingly, the self-supervised methods [19] reach a competitive performance to those that rely on the availability of dense annotations (like semantic segmentation and ground truth depth) for supervised representation learning [58, 74]. Furthermore, methods that learn a curiosity-based representation (CRL [22]) or via auxiliary tasks and RL (SplitNet [30]) do not transfer as well as the SSL and SL methods. Additionally, compared to the strong DD-PPO [65] approach which was trained on the same data as our policy but for the PointNav task, our model achieves substantial gains in success rate (from +5% and up to +14%) across all tasks. This indicates that our semantic search policy is much better suited to transfer to diverse downstream tasks compared to the PointNav policy. Moreover, when looking at the test performance over the course of training for the best transfer methods compared to ours (Fig. 3), we notice that our approach has a much higher start and improves faster to better performance. Our model reaches the top performance of the best competitor up to 12.5× faster.
Zero-Shot Experience Learning  A unique feature of our approach is its ability to perform the downstream task without receiving any new experiences from it. Our model shows excellent performance under the challenging ZSEL setting. Our ZSEL model outperforms the Task Expert in 4 out of 5 of the tasks despite receiving zero new experience on the target, and even after training the Task Expert for up to 20 million steps (Ours-ZSEL in Table 3).

In addition, we see in Fig. 3 that the majority of transfer learning models struggle to reach our ZSEL performance. Note that our model does not have any advantages in terms of architecture, which is shared with the rest of the models. Thus, the high ZSEL performance is attributable to our modular transfer approach. In ObjectNav and RoomNav, the best competitor requires between 2 to 16 million steps to reach our ZSEL performance, and with the exception of ObjectNav-Audio the competitors show little improvement over that level. In ViewNav, we notice that none of the baselines are capable of reaching our ZSEL level. This can be attributed to the challenging goal modality where estimating distances for successful stopping is difficult, and to the close proximity of this task to the source ImageNav task that our semantic policy is most familiar with.

Modular Transfer Ablation  Fig. 4 (left) shows a modular ablation of our approach on the ObjectNav-Label task. Transferring individual modules separately has mixed impact on performance. While transferring \( f_G \) and \( \pi \) only does not improve over the ‘No Transfer’ case, \( f_O \) leads to positive transfer effect. This is expected since in this model \( f_O \) is a deep CNN with the largest portion of parameters. Having a good initialization of this component is beneficial. Nonetheless, when combining the modules together with our plug and play modular approach we see substantial gains. Our full model demonstrates the best performance and enables ZSEL, thus validating our contributions.

Scalability  We evaluate our model’s ability to scale in terms of experience gathered on the source task. We find a strong correlation between the experience gathered on the source task and the ZSEL performance on downstream ones. As our semantic search policy receives more experience on the source task (ImageNav), its ZSEL performance on the target task gets better (Fig. 4 right). This is important since our source task requires no annotations and can be easily scaled to more scenes and large datasets. For an analysis of our model in terms of the used sensors, see Supp.

Long-Term Task Expert Training  We saw above that our model scales well and its transfer improves when more experience is gathered on the source task. However, does a task expert become competitive if it simply gets longer training on the target task? How long does that model take to catch up with our approach? To find out, we train the Task Expert on each of the target tasks for up to 500M steps. Fig. 5 shows the results. The Task Expert requires on average more than 22M steps on ObjectNav and RoomNav, and up to 416M on ViewNav (in total 507M steps over the 5 tasks) to reach our ZSEL performance. It never reaches our model’s top performance when our model is finetuned on the target task. Moreover, our model reaches the best performance of the Task Expert 34.7\( \times \) faster. The Task Expert needs task-specific experience with task-specific annotations, which can be expensive and limits the available training data. In contrast, our model learns in the source task using more diverse goals that can be sampled randomly from the (unannotated) scene, thus scaling more effectively.

Additional Results and Discussions  Please see Supp for qualitative results, an analysis of failure cases, and a discussion of limitations and the societal impact of our approach.

5. Conclusion

We introduce a plug&play modular transfer learning approach that provides a unified model for a diverse set of semantic visual navigation tasks with different goal modalities. Our semantic search policy outperforms the SoTA in the source task of image-goal navigation, as well as the SoTA in transfer learning for visual navigation by a significant margin. Furthermore, our model is able to perform new tasks effectively with zero-shot experience—to our knowledge, a completely new functionality for visual navigation. This is a stepping stone for future work, especially for tasks with high-cost training data. Being able to do ZSEL and learn from few experiences is a crucial skill for an agent in open-world and lifelong learning settings.

Acknowledgements:  UT Austin is supported in part by DARPA L2M, the UT Austin IFML NSF AI Institute, and the FRL Cog Sci Consortium. K.G is paid as a Research Scientist by Meta AI. Thanks to Lina Mezghani for providing access to data and code.
References


[38] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009. 2, 3


