Situational Fusion of Visual Representation for Visual Navigation

William B. Shen[1] Danfei Xu[1] Yuke Zhu[1] Leonidas J. Guibas[1,2] Li Fei-Fei[1] Silvio Savarese[1]
[1]Stanford University, [2]Facebook AI Research
{willshen,danfei,yukez,guibas,feifeili,ssilvio}@cs.stanford.edu

Abstract

A complex visual navigation task puts an agent in different situations which call for a diverse range of visual perception abilities. For example, to "go to the nearest chair", the agent might need to identify a chair in a living room using semantics, follow along a hallway using vanishing point cues, and avoid obstacles using depth. Therefore, utilizing the appropriate visual perception abilities based on a situational understanding of the visual environment can empower these navigation models in unseen visual environments. We propose to train an agent to fuse a large set of visual representations that correspond to diverse visual perception abilities. To fully utilize each representation, we develop an action-level representation fusion scheme, which predicts an action candidate from each representation and adaptively consolidate these action candidates into the final action. Furthermore, we employ a data-driven inter-task affinity regularization to reduce redundancies and improve generalization. Our approach leads to a significantly improved performance in novel environments over ImageNet-pretrained baseline and other fusion methods.

1. Introduction

Assistive robots that can efficiently navigate an everyday home require an extensive repertoire of visual perception abilities. To "find the nearest cup", the robot needs a situational combination of different perception abilities. It needs to recognize a cup using object detection, and if no cup is present, identify and navigate to a room that may contain cups using a combination of scene semantic knowledge and geometric cues such as vanishing point and depth. Performing such a complex task in *any home* that may have completely different spatial layouts and appearances demands not only the generalizability of the individual visual perception modules but also the situational combination thereof.

Recently, deep reinforcement learning has made promising progress in goal-directed visual navigation with end-to-end learning [21, 41]. The guiding principle of imposing *minimal structure* to the learning algorithm and the practically unlimited synthetic training data have warranted the wide applicability of these methods. However, obviating structures and prior knowledge about visual perception can often lead to solutions that overfit to an environment. In fact, such success often reckons on training and evaluating in identical or similar environments [21, 41]. Being able to perform complex navigation tasks and generalize to environments with drastically varying appearances as aforementioned is still beyond reach.

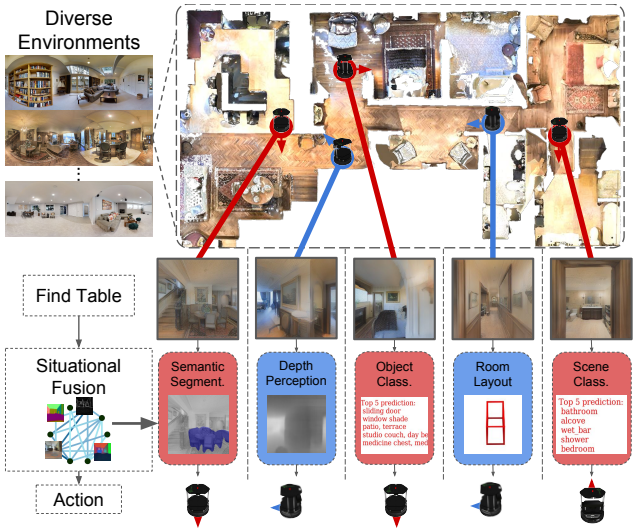


Figure 1: (Top) The visual navigation task requires an agent to handle large diversity in real-world environments. (Bottom) Instead of using a black-box model, we propose to adaptively fuse *visual representations* from a diverse set of vision tasks to better generalize to new environments.

Meanwhile, *visual representations* extracted from a wide range of vision tasks [6, 19] have been greatly effective in generalizing towards new tasks and new domains. It has been a common practice to reuse visual representations trained for standard vision tasks, such as image classification [20], in new datasets and new problems [8, 33]. In this work, we endow visual navigation models with structures and priors based on a diverse set of visual representations. This leads to stronger generalization without losing the wide applicability of end-to-end learning. Our idea is

motivated by the recent work of Taskonomy [38], which introduced a computational framework for studying the similarity and transferability among vision tasks. Their data-driven result shows that effective transfer of representation can happen between tasks of high affinity, which can be used to identify redundancies among tasks and significantly reduce total supervision. The repertoire of visual representations and their affinity measures from Taskonomy offer a basis for our model to learn generalizable navigation policies that transfer to unseen environments.

Thus, in order to harness visual representations for visual navigation, the primary challenge is to devise a scheme that adaptively fuses the representations based on a situational understanding of the task, while not overfitting to spurious dependencies among visual representations. To this end, we introduce an end-to-end approach that fuses visual representations at the action level and uses task affinity as regularization (see Fig. 1).We learn an action predictor for each representation, and then combine the action predictions from all representations into the final action output. We also use the data-driven task affinity discovered in Taskonomy as a source of regularization to penalize selection of redundant representations. This method ensures that the fused representation strikes a balance between being informative of the trained task and being generalizable towards new scenarios.

Prior works that addressed generalization of navigation policy [14, 27] have typically focused on simulation-to-real transfer for low-level motion control tasks. We instead evaluate our approach in visual navigation tasks with higher complexity in the Gibson simulated environments [37], which were shown to transfer to the real world without further supervision [24, 17]. Our policy takes as input RGB images and produces the action command for the robot. Our results indicate that the use of representations has led to substantially better generalization for a high-level navigation task in unseen environments, and fusion at the action level leads to better generalization and robustness towards noise and subsystem failure. Furthermore, the inter-task affinity regularization promotes the fusion method to select more complementary and less redundant representations.

To summarize, our contributions are twofold: 1) we propose to use visual representations from a diverse set of vision tasks as a prior to learn generalizable policies for visual navigation, and 2) we develop action-level representation fusion and regularization techniques to achieve strong zero-shot generalization results in unseen environments.

2. Related Work

Representation Learning. Prior work has developed various methods to learn visual representations with deep neural networks using supervised learning [6], unsupervised learning [7, 29], and weakly supervised learning [16] objectives. Motivated by the success of using deep features in transfer learning, a series of work has developed visualization and analysis techniques to understand the characteristics of deep features [22, 39] and the factors that affect the efficacy of transfer [15]. Taskonomy [38] has recently demonstrated that representations learned on 25 distinct vision tasks can transfer between similar tasks. However, it has focused on static image-based tasks. In contrast, we study how to leverage these diverse visual representations to effectively learn generalizable policies for visual navigation.

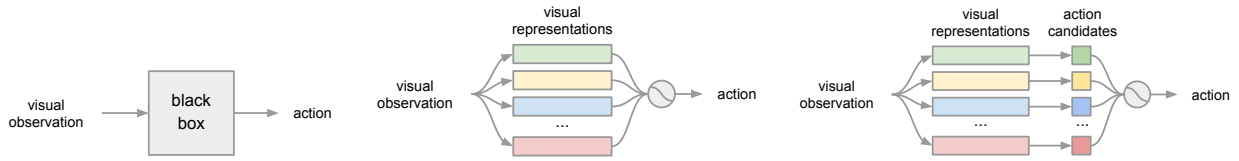
Visual Navigation. Vision-based navigation for mobile robots has been widely studied in classic robotics literature [1, 35]. The recent boom of deep learning has driven a new wave of development of visual navigation methods that employ the representational power of deep networks. Deep reinforcement learning [25, 26, 28, 41] has been successfully applied to mapless navigation, eliminating the need for an explicit environment map. Other methods have adopted planning modules in the learning pipeline [10], or tackled the navigation task along with other semantic tasks, such as language grounding [13] or visual question answering [4, 9]. Most of the prior work has focused on learning these navigation models end-to-end from scratch rather than reusing visual representations learned from other vision tasks. We demonstrate that our model can effectively leverage these representations to achieve stronger generalization when navigating in an unseen environment.

Generalization in Policy Learning. A series of methods have been developed to improve the generalization of policy learning towards novel visual observations [36], noisy environment dynamics [23, 30], and new task instances [31, 41]. Synthetic data has been harnessed as a training source to empower the training of the data-hungry deep network policies. Accordingly, several works have introduced new techniques to close the reality gap, allowing these policies to generalize from simulation to the real world [14, 27, 40]. In this work, we also use simulated data to train our models. However, our simulator of choice, Gibson [37], employs 3D captures of real-world scenes and domain-adapted simulation, which has been shown effective in deploying the simulation-trained policies directly in the real world [17, 24]. We focus on improving the generalization of our models across visual scenes with novel ways of fusing the diverse set of visual representations.

3. Situational Visual Navigation Model

Our goal is to learn visual navigation policies that generalize better to unseen environments by enabling the agent to adaptively fuse *visual representations* which are trained on a diverse set of vision tasks [38]. However, naïvely fusing representations results in overfitting (Table 1), and it is not robust against noise and subsystem failure (Fig. 7).

To address overfitting and to increase model robustness,



(a) black-box model (b) feature-level fusion model (c) action-level fusion model

Figure 2: Three schemes of learning neural network policies from visual perception to action, including end-to-end learning with black-box neural networks, and representation fusion methods at both feature and action levels.

we propose to combine the representations at the action level (Sec. 3.3.1), which ensures that each representation is individually trained to make meaningful prediction for the overall task. We also introduce a novel regularization over the fusion weights based on inter-task affinity (Sec. 3.3.2). It reduces the co-selection of redundant representations and diversifies the use of representations based on the characteristics of each state. Results show that the synergy of action-level fusion and inter-task affinity regularization (Fig. 3) yields big performance gain over baselines.

3.1. Problem Formulation

We focus on the task of visual navigation in indoor environments. The agent perceives the environment through its on-board RGB camera [37]. Following the setup in prior work [10, 41], the actions are defined as high-level commands $a_{x,\theta}$, where θ is the turning angle and x is the stepping distance. We assume that the environments are discretized into a regular octagonal grid, where the agent uses the high-level actions to traverse on the grid.

We formulate the learning problem as follows. At the beginning of an episode, the agent is randomly spawned at location $p_0 = (x_0, y_0)$ in an environment $\mathcal{E}$. At each step, the agent receives a visual observation in the form of RGB image $o_t = \mathcal{O}(p_t, \mathcal{E})$, where $\mathcal{O}$ is a function which returns the current images at location p_t in $\mathcal{E}$. From each visual observation, a set of visual representations $\mathbf{r}_t(o_t)$ can be computed from deep network models trained on a diverse set of vision tasks, where $\mathbf{r}_t(o_t) = \{\mathbf{r}_t^1, \mathbf{r}_t^2, ...\}$. We want to learn a closed-loop navigation policy $\pi(a_t|o_t, \mathbf{r}_t)$, parameterized by neural networks, to map the RGB image o_t and their visual representations $\mathbf{r}_t$ to the action a_t that commands the robot to navigate in the environment. For each task, we specify a Boolean function that determines if the current location p_t satisfies the goal condition. Our objective is to learn an optimal policy π^* which reaches a goal location from its current location in minimum number of steps.

3.2. Representations from Diverse Vision Tasks

The visual navigation task requires a variety of visual perception abilities, thus demanding visual representations learned from a diverse set of vision tasks. We leverage the representations from Taskonomy [38], which trains deep network models for 25 distinct computer vision tasks at multiple levels of abstraction. For each task, a neural network model is trained with supervised learning to map the input RGB image into a compact representation $\mathbf{r} \in \mathbb{R}^{16\times16\times8}$, which capsules the information required to solve the task. It has been shown that these representations, being compact in size, are easily transferable to other visual understanding tasks. The compactness and richness of these representations make them appealing for representing the visual perception abilities required in interactive visual tasks. Therefore, we use these 25 representations as a basis to learn our vision-based navigation policies.

Taskonomy also introduced a data-driven affinity score between each pair of the 25 vision tasks. The affinity scores estimate the correlation and redundancy of the representations across these tasks. We demonstrate in Sec. 3.3.2 that we can incorporate such affinity of representations into the process of situational fusion. This encourages the model to make a more balanced selection of representations and reduce overfitting to redundancies.

3.3. Situational Representation Fusion

A naïve approach to learn the navigation policies is to consider deep network as a black-box model and train it with end-to-end learning (Fig. 2a). While this approach can directly optimize the learning objective from pixel to control, it tends to capture spurious dependencies from limited training data, hindering its generalization ability. Prior work [38] has proposed to leverage Taskonomy representations via a simple concatenation of $\mathbf{r}_t(o_t)$. However, it offers marginal performance gain over the black-box model (Table 1). We hypothesize that this is due to the lack of situational use of representations.

Our key intuition is that different stages of the visual navigation task would require different visual perception skills. For instance, localizing the target object would require semantic understanding, and avoiding obstacles would require geometric reasoning. To satisfy such requirements, the model has to develop a situational understanding of its current perception for decision making. Thus, we introduce a situational-fusion module that adaptively weighs and combines the representations based on the current state.

A widely used paradigm of combining representa-

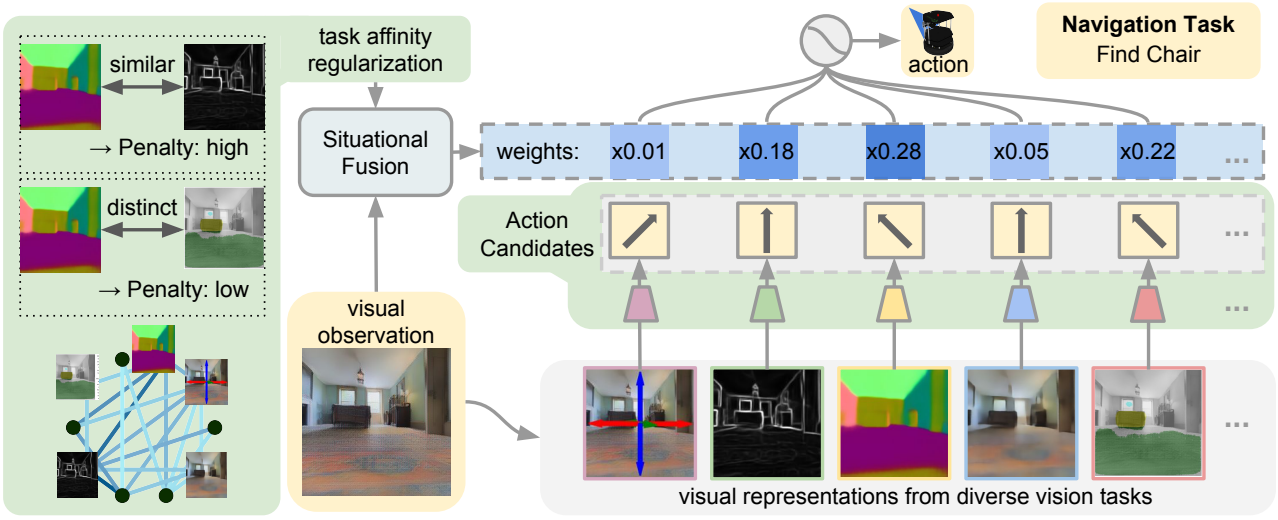


Figure 3: The agent receives an RGB image and its corresponding visual representations. We use a situational-fusion network to adaptively weigh and combine the representations at the action level based on the current observation. The fusion weight is regularized by inter-task affinity to encourage a more balanced selection of the representations for better generalization.

tions [5, 32] is to use fused representations as joint input to neural network layers (i.e., feature-level fusion illustrated in Fig. 2b). However, as prior work [32] pointed out, such fusion method tends to bias towards a few dominating representations and is prone to poor generalization performance. This leads to the two challenges of overfitting and lack of robustness, as highlighted at the beginning of this section. We introduce two effective techniques to deal with the challenges as follows.

3.3.1 Fusion at the Action Level

As described at the beginning of this section, feature-level fusion faces the challenge of overfitting, and lack of robustness towards noise and subsystem failure.

In robotics, to perform a complex task, compositionality of behaviors [2, 34] promotes to learn a valid behavior for every sub-module, which ensures that all sub-modules are well trained. Then the simpler behaviors are composed at decision-output level into more complex behaviors. Since each sub-module develops a valid behavior, the overall system also becomes more robust towards sub-module errors.

Inspired by this approach, we propose an alternative way of fusing the representations at the action level (Fig. 2c). We learn a valid action-prediction model out of each representation, which ensures that all representations are well trained (Fig. 3). Formally, at any time step t, as the agent receives RGB image input o_t, it uses a situational-fusion network f to output an n-dimensional vector $\mathbf{h}_t = f(\mathbf{o}_t)$ for the n representations. $\mathbf{h}_t$ is then normalized with a softmax function to obtain the fusion weight $\mathbf{g}_t = \texttt{softmax}(\mathbf{h}_t)$. We use pretrained Taskonomy modules to compute the n representations for the image $\{\mathbf{r}_t^1, \mathbf{r}_t^2, \ldots, \mathbf{r}_t^n\}$. For each representation $\mathbf{r}_t^i$, an action-prediction module $\pi'_{i,\theta}(a|\mathbf{r}_t^i)$ is independently trained and produces an action candidate $\tilde{a}_t^i$, thus not suffering from under-training. Each action candidate can be independently taken as a final action for execution, but better decision can be made by using situational-fusion weights $\mathbf{g_t}$ to reweigh and combine the action candidates into the final action output a_t:

$$a_t = \sum_{i=1\ldots n} \mathbf{g}_t^i \, \pi'_{i,\theta}(\mathbf{r}_t^i), \; \mathbf{g}_t = \texttt{softmax}(f(\mathbf{o}_t))$$

We observe that using this new fusion scheme significantly improves generalization performance over naïve fusion scheme (Table 1), and the action candidate of each branch achieves reasonable individual performance (Fig. 6). At the same time, it also offers more robustness towards noise and sub-system failure (Sec. 4.3). However, although individual branches are well developed in the action-level fusion scheme, we observe that the fusion weights still show strong bias towards a few dominating action candidates during inference. In the following section, we propose a novel regularization loss using inter-task affinity.

3.3.2 Inter-task Affinity Regularization

The bias of fusion weights towards a few dominating representations makes the overall model prone to overfitting. To deal with the problem, prior work [32] adopted a load-balancing loss (LBL) term to regularize on coefficients of variation $\mathbf{L}_{\texttt{LBL}} = \texttt{CV}(\mathbf{g}_t)$, where $\texttt{CV}$ stands for the coefficients of variation operator. However, LBL loss promotes uniform fusion weights across all representations, and we observe that adding this regularizing term alone to our fusion model offers limited to no performance improvement for both feature-level and action-level fusion. A clear shortcoming of this technique is that it fails to take into account the correlations among representations. High correla-

tion between two representations means that choosing both leads to more information redundancy, which can affect the model's generalization ability. Therefore, we further propose to design a loss that leverages correlations of representations as a prior to minimize information redundancy and balance fusion weights. Taskonomy [38] uses a data-driven approach to measure pairwise task-affinity by calculating the performance gain from using one task's representation to transfer-learn the other task. We use Taskonomy's data-driven pairwise task affinity as a surrogate for the correlations among different visual task representations.

Formally, for each time step t and fusion weight $\mathbf{g}_t \in \mathbb{R}^n$, we want to encourage large weights in $\mathbf{g}_t^i, \mathbf{g}_t^j$ if task i and j have low task affinity $\texttt{aff}(i,j)$, and smaller weight if task affinity is high. To achieve this, we inject a regularizer for task affinity by adding a bilinear loss term [11] between fusion-weight vector and task-affinity matrix (Fig. 3):

$$\mathbf{L}_g(\mathbf{g}_t) = \mathbf{g}_t^T \mathbf{F} \mathbf{g}_t, \ \mathbf{F}_{i,j} = \texttt{aff}(i,j) \tag{1}$$

For every pair of tasks, we measure the product of the task affinity $\texttt{aff}(i,j)$ and its corresponding fusion weights $\mathbf{g}_t^i, \mathbf{g}_t^j$, and calculate $\mathbf{L}_g(\mathbf{g}_t)$ as sum of all such products. For example, *occlusion edge detection* and *surface-normal estimation* are two tasks with high affinity. If the model puts large weights on both tasks in $\mathbf{g}_t$, then the product between their task affinity and fusion weights will be large, which leads to a larger $\mathbf{L}_g(\mathbf{g}_t)$ and the selection will be penalized. Thus, regularizing on this loss encourages the situational-fusion network to reduce information redundancy and balance fusion weights.

4. Experiments

We evaluate our proposed methods of situational fusion in a set of visual navigation tasks. Throughout the experiments, we measure the generalization aspect of different navigation policies, and examine the effectiveness of action-level fusion and the inter-task affinity regularization.

4.1. Experiment Setup

Experimental Testbed. We conduct experiments in GibsonEnv [37] rendering of Matterport3D assets [3], which were captured with real-world 3D scans and labeled with semantic ground truth. We use a diverse set of 62 simulated indoor environments that contain our objects of interest. We focus on the high-level visual navigation planning and map the navigation locations to an octagonal grid. At each time step, the agent receives eight RGB images in the octagonal directions obtained from its on-board 360-degree camera. The action space consists of eight actions that step along the eight directions on the octagonal grid and a stop action. Following [10], we preprocess all traversable locations on the octagonal grid and generate a directed graph $\mathcal{G}_{x,\theta}$ that connects these locations. This enables us to acquire the supervision of optimal actions to train our policies through shortest-path algorithms.

Task Setups. We follow the formulation of semantic navigation tasks [10]. The agent is commanded to "go to the closest X," where $X \in \{chair, table, bed, door\}$. We use the ground truth object annotation from [3] to label nodes in the graph $\mathcal{G}_{x,\theta}$ with object categories, and define the optimal action as stepping towards the closest instance of the object class. To ensure the plausibility of finding a solution, we ensure the agent to start in a room where at least one instance of the specified object class exists. In our tasks, the maximum shortest-path distance between the agent's starting location and a target object is 32 steps, and the minimum is 6. This setup requires the agent to learn object appearances through algorithmic supervision and find the same object class (different instances) in novel test environments.

Evaluation Protocol. We follow the train/test procedures used in prior work [10]. For each task, we use on average 28 environments for training and 14 for testing. An episode is judged to be successfully completed if the agent, given a maximum of 39 steps, ends in a location within 3 steps from the specified object. During testing, we randomly sample a fixed set of 1024 starting locations in the test environments, and report the success rate of each model.

Baselines. We first compare our method with three baselines which do not use situational fusion:

- `Random`: a random walk agent that takes a uniformly random action at each time step;
- `ResNet`: a black-box deep neural network that maps from raw pixel inputs to action labels (see Fig. 2a). We use ResNet-50 model [12] as in Taskonomy [38] with pretrained-weights from ImageNet[6]. Same data augmentation as [12] is applied.
- `Concat`: directly concatenating all representations [38].

We then compare our method with naïve situational-fusion baselines and prior work [32]:

- `Feature-level Fusion`: naïve situational fusion of representations at feature-level (Sec. 3.3).
- `LBL`: adding load-balance loss [32], examined for both feature-level and action-level fusion models.

We report two additional baselines for action-level fusion:

- `Maj.`: we perform majority voting among the action candidates and select the top one. This corresponds to a simplified version of our action-fusion method where the fusion weight is set to uniform.
- `Top k`: to understand the impact of increasing number of representations, we perform majority voting on the actions from the branches that have the k-highest individual feature success rates (Fig. 6).

Proposed Models. We examine our proposed extensions to feature-level fusion. We explore the effectiveness of fusion at action-level (Sec. 3.3.1), which combines a set of

Tasks	Random	ResNet [12]	Concat	Feature-level Fusion				Action-level Fusion						
				-	LBL [32]	T.Aff [38]	Both	Top 1	Top 5	Maj.	-	LBL [32]	T.Aff [38]	Both
Bed	2.0	39.6	33.6	30.1	32.4	34.8	44.0	55.0	53.1	45.3	48.8	50.0	48.8	45.7
Chair	4.4	18.5	21.7	13.1	15.7	16.3	21.1	19.9	24.6	30.8	28.9	31.2	36.3	34.3
Table	3.8	17.3	17.1	17.7	21.4	19.7	26.4	19.5	24.2	37.8	28.9	25.7	35.9	39.8
Door	2.1	8.3	29.8	31.3	32.5	33.1	33.2	42.5	54.6	50.7	54.6	52.3	55.8	56.2
Avg.	3.0	20.9	25.6	23.1	25.5	26.0	**31.2**	34.2	39.1	41.1	40.3	39.8	**44.2**	44.0

Table 1: Quantitative evaluation (success rate) of visual navigation policies in unseen environments.

action candidates predicted from individual representations (`action-level fusion`). Then we investigate inter-task affinity regularization (Sec. 3.3.2) that discourages redundancies in fusion weights (`T.Aff`).

All models were trained using ADAM [18] to optimize for the loss function and trained for 16K iterations with batch size of 256 (64 for `ResNet` baseline). We decay the learning rate by a factor of 10 every 5k iterations.

4.2. Quantitative Evaluation

Table 1 compares among baselines and variations of our proposed model. The table consists of five main columns: random agents, ResNet baselines, concatenating representations, feature-level fusion and action-level fusion. We measure each model's performance on all four navigation tasks and report the average across tasks. The numbers are evaluated in test environments unseen during training.

First, we observe a large performance gap between black-box methods that train on raw pixels and our proposed methods that situationally fuse visual representations at action-level with inter-task affinity regularization. The method achieves a 2× higher success rate than the state-of-the-art pretrained ResNet model [12]. This indicates that correctly leveraging visual representations has a significant effect in improving generalization of the learned policy.

Second, we see that naïvely using visual representations results in limited performance increase. Directly concatenating 25 different visual representations only results in 5% performance increase over the ResNet model, and a vanilla feature-level fusion model performs similarly. As mentioned in Sec. 3, two challenges (overfitting and lack of robustness) need to be solved. We can see that adding load-balancing loss (`LBL`) [32] alone only offers marginal help.

Our proposed action-level fusion explicitly learns an action-prediction model for each representation. Since the load-balancing problem in training is explicitly handled, results show that action-level fusion significantly outperforms `LBL` loss and that adding `LBL` loss onto action-level fusion brings no further gain. As for comparing with feature-level fusion, action-level fusion shows superior performance. While both schemes have similar training performances, action-level fusion performs much better for testing (Fig. 4). We hypothesize that action-level fusion suffers less overfitting thanks to its separate handling of each representation. Fusion of the action candidates acts as a model ensemble, which can effectively factor in each representation's contribution in decision making. This hypothesis is further backed by the competitive results of the Majority Voting baseline, which combines the actions with uniform weights.

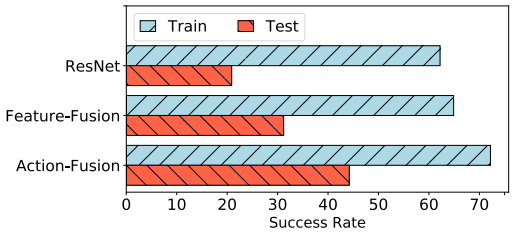


Figure 4: Training and testing performance comparison. Testing scenes are previously unseen environments. Although models achieve similar levels of performance on training scenes, our action-level fusion model generalize significantly better to unseen test environments than baselines.

Finally, we see that adding an inter-task affinity regularization (`T.Aff`) improves the agent's performance in both fusion schemes. `T.Aff` outperforms `LBL` loss in both feature-level and action-level fusion. For feature-level fusion, since `T.Aff` is not designed for dealing with representation under-training, combining it with `LBL` loss gives a better performance. For action-level fusion, which is more effective towards representation under-training than `LBL`, incorporating `T.Aff` has achieved the best performance and outperformed the Majority Voting baseline by 3%. The effect is more significant for tasks of finding table and chair, where the performance boost is over 7%. Results show that the task affinity regularization is able to reduce spurious dependencies and encourage more balanced selection.

4.3. Model Analysis

Analysis of Fusion Weights: We conducted qualitative and quantitative studies on the distribution patterns of the situational-fusion weights, results shown in Fig. 5.

In the first test, we group the representations into three domains: 2D (e.g. autoencoder), 3D (e.g. depth estimation) and semantics (e.g. semantic segmentation). For each location in our environments, we record the fusion weights of our action-level fusion model. We group the weights based on the representation domains, perform normaliza-

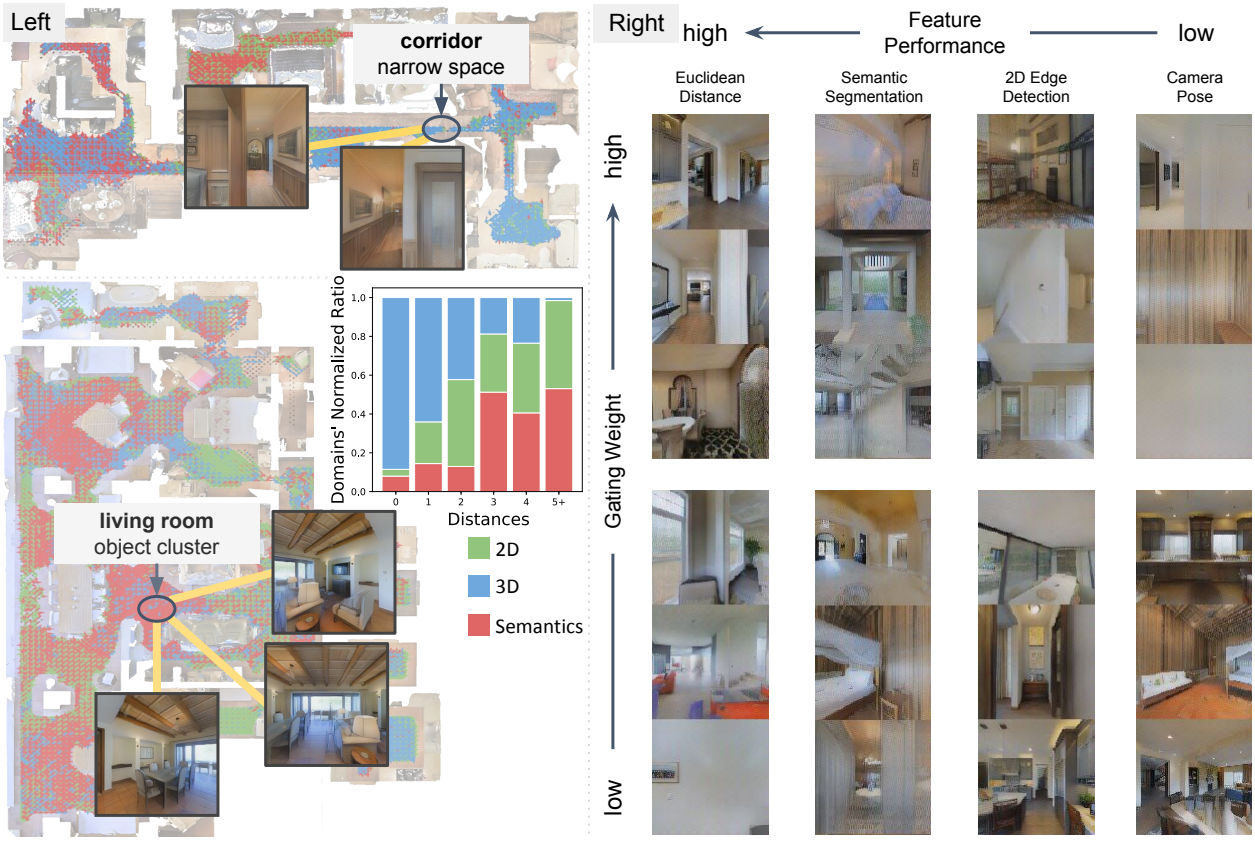


Figure 5: (**Left**) fusion weight heat-map of different representation domains on two example environments, and a quantitative bar chart of distribution over all testing scenes. 3D representations receive higher weights in narrow space like corridor, while semantic representations become more dominant in object clustered space. (**Right**) representation's top and bottom activation image based on its fusion weights. Representations with better individual performances were higher weighted in complex scenes and lower in simple scenes; the opposite is true for worse representations.

tion, and visualize the top domain, shown in Fig. 5 left. The color code is **red** for semantic tasks, **green** for 2D, and **blue** for 3D. 3D representations are the most activated in narrow spaces such as corridor, where the main challenge of the agent is to go through and not collide. In open area with complex object clusters, weight skews more to red, showing more involvement of semantic representations. For quantitative analysis, we use each position's distance (in steps) towards the closest obstacle to its sides as a surrogate for surrounding's openness. For each representation domain, we compute the percentage of positions with highest weights in such domain, and then normalize across distance values. The results are shown in Fig. 5's bar chart, with Y-axis showing distance and X showing ratio. As surrounding becomes narrower, fusion shifts more from **semantics** to **3D**.

In the second test, we examine how weights are distributed between representation branches with different individual performances (see Fig. 6). We sample 100 images from each test environment, and record the fusion weights across samples. Then, for each representation, we select the top 4 images with the highest corresponding fusion weights as well as the lowest bottom 4 images, shown in the right part of Fig. 5. Due to space limit, we spread out only 4 representations of different individual-performance levels.

The columns in Fig. 5 are sorted from left to right according to individual representation performances measured in Fig. 6. As we can see, high-performing representations are highly activated when facing more complex scenes with lots of objects and varying spatial layout (Fig. 5 top left). They have relatively lower weights when the view becomes more plain (Fig. 5 bottom left). Low-performing representations, like 2D edge detection or camera pose estimation, demonstrate the opposite. This suggests that our fusion mechanism learns to put higher reliance on high-performing representations when facing hard decisions (more complex observations), and incorporate low-performing representations to hedge risks when facing simpler decisions.

Analysis of Individual Representations: To investigate the contribution of each visual representation in visual navigation, we quantitatively measure the success rate of directly executing the action candidates from each branch for our action-level fusion models in Fig. 6.

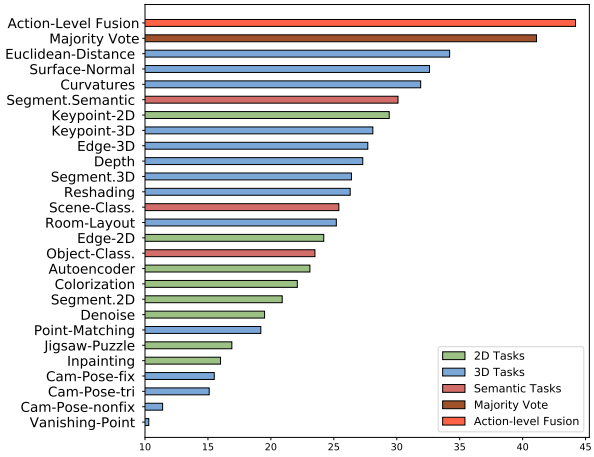


Figure 6: Comparison among our action-level fusion model, majority-voting baseline and each individual representation's performance. Individual branches are color-coded according to task types, with green for 2D tasks, blue 3D, and red semantics.

As we can see, the top three best-performing representations are all 3D geometric tasks. *Euclidean-Distance*, being the top-performing representation, should constantly be consulted so that the agent does not collide into object. *Surface-Normal Estimation*, proved to be one of the best representation to transfer out [38], clearly demonstrates its usefulness in navigation tasks. Semantic representations, especially semantic segmentation, ranks relatively high. For our navigation tasks, semantic information is important in an agent's success of locating the target objects.

Low-level vision tasks, such as vanishing-point estimation and camera-pose estimation, perform poorly as an individual model. These representations have very high abstraction level, and it is hard for the agent to read out scene layout information necessary for complex behaviors.

Analysis of Model Robustness: Robustness to unexpected scenarios is very important for a navigation agent. At any moment, a representation might show out-of-distribution noise, and in extreme cases, there might be sub-system failures (such as bugs or attacks) that prevent the agent from accessing its complete representation set. If the agent is unequipped to handle noise or absence of representation, there might be severe consequences.

Action-level fusion provides a robust way to deal with such challenges. Since each representation individually produces an action output, the downstream effect of missing representations is kept to a minimum. On the other hand, any defect to representation will propagate to later layers of the policy network for feature-level fusion. To measure robustness to representation loss, we conduct two tests for the top model of feature-level fusion and of action-level fusion. In both tests, we randomly select a set of representations to be dropped at each step.

For the first test (Fig. 7 left), we assume that the loss of

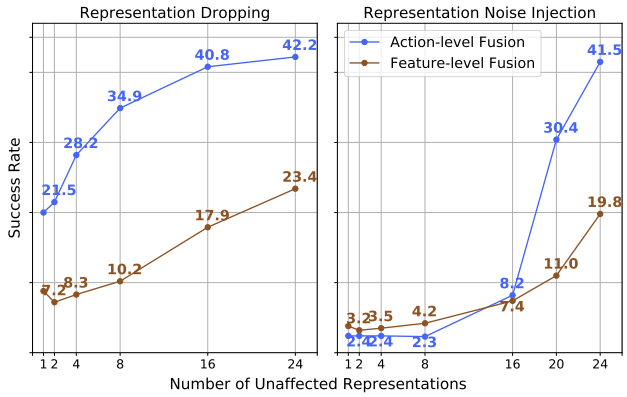


Figure 7: **Robustness Analysis**: The left figure shows how performance varies when a number of representations are dropped. The right figure shows how performance varies when a number of representations are set to zero. Horizontal axis shows how many representations remain unaffected.

representation is noticed by the agent, and it skips the given representation branch by setting the lost representations' fusion weights to zero and normalizing the remaining values. As we see, action-level fusion handles representation loss much better than feature-level fusion.

For the second test (Fig. 7 right), we randomly replace a set of representations with noise (setting all values of the representations to zero), and the agent treats the affected representation as usual. This is a more challenging setup since the agent might put high weights on the affected representations. We can see that action-level fusion is more robust than feature-level fusion, and is able to have meaningful performance when a random set of 5 representations is affected at each time step.

5. Conclusion

We explored the effectiveness of situationally fusing visual representations from vision tasks to improve zero-shot generalization of an interactive agent in visual navigation tasks. We proposed two novel extensions to fusing representations: action-level fusion and inter-task affinity regularization. Our results suggested that a combination of the two extensions led to better generalization and enhanced robustness of the navigation agent. In a broader scope, these promising results shed light on how to build intelligent systems that can effectively leverage internal representations from other tasks to learn new behaviors. One of the possible future directions is to apply similar principles of representation fusion to other interactive domains, such as active perception and visuomotor learning.

Acknowledgement: We thank Andrey Kurenkov and Ajay Mandlekar for helpful comments. Toyota Research Institute (TRI) provided funds to assist authors with their research, but this article solely reflects the opinions and conclusions of its authors and not TRI or any other Toyota entity.

References

[1] Francisco Bonin-Font, Alberto Ortiz, and Gabriel Oliver. Visual navigation for mobile robots: A survey. *Journal of intelligent and robotic systems*, 53(3):263–296, 2008. 2

[2] Rodney Brooks. A robust layered control system for a mobile robot. *IEEE journal on robotics and automation*, 2(1):14–23, 1986. 4

[3] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017. 5

[4] Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Embodied question answering. *arXiv preprint arXiv:1711.11543*, 2017. 2

[5] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 933–941. JMLR. org, 2017. 4

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. 2009. 1, 2, 5

[7] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Unsupervised visual representation learning by context prediction. In *International Conference on Computer Vision (ICCV)*, 2015. 2

[8] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 1

[9] Daniel Gordon, Aniruddha Kembhavi, Mohammad Rastegari, Joseph Redmon, Dieter Fox, and Ali Farhadi. Iqa: Visual question answering in interactive environments. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018. 2

[10] Saurabh Gupta, James Davidson, Sergey Levine, Rahul Sukthankar, and Jitendra Malik. Cognitive mapping and planning for visual navigation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2616–2625, 2017. 2, 3, 5

[11] Trevor Hastie and Robert Tibshirani. Efficient quadratic regularization for expression arrays. *Biostatistics*, 5(3):329–340, 2004. 5

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 5, 6

[13] Karl Moritz Hermann, Felix Hill, Simon Green, Fumin Wang, Ryan Faulkner, Hubert Soyer, David Szepesvari, Wojtek Czarnecki, Max Jaderberg, Denis Teplyashin, et al. Grounded language learning in a simulated 3d world. *arXiv preprint arXiv:1706.06551*, 2017. 2

[14] Zhang-Wei Hong, Chen Yu-Ming, Shih-Yang Su, Tzu-Yun Shann, Yi-Hsiang Chang, Hsuan-Kung Yang, Brian Hsi-Lin Ho, Chih-Chieh Tu, Yueh-Chuan Chang, Tsu-Ching Hsiao, et al. Virtual-to-real: Learning to control in visual semantic segmentation. *arXiv preprint arXiv:1802.00285*, 2018. 2

[15] Minyoung Huh, Pulkit Agrawal, and Alexei A Efros. What makes imagenet good for transfer learning? *arXiv preprint arXiv:1608.08614*, 2016. 2

[16] Armand Joulin, Laurens van der Maaten, Allan Jabri, and Nicolas Vasilache. Learning visual features from large weakly supervised data. In *European Conference on Computer Vision*, pages 67–84, 2016. 2

[17] Katie Kang, Suneel Belkhale, Gregory Kahn, Pieter Abbeel, and Sergey Levine. Generalization through simulation: Integrating simulated and real data into deep reinforcement learning for vision-based autonomous flight. *arXiv preprint arXiv:1902.03701*, 2019. 2

[18] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[19] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123:32–73, 2016. 1

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 1

[21] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016. 1

[22] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2

[23] Ajay Mandlekar, Yuke Zhu, Animesh Garg, Li Fei-Fei, and Silvio Savarese. Adversarially robust policy learning: Active construction of physically-plausible perturbations. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, pages 3932–3939. IEEE, 2017. 2

[24] Xiangyun Meng, Nathan Ratliff, Yu Xiang, and Dieter Fox. Neural autonomous navigation with riemannian motion policy. In *IEEE International Conference on Robotics and Automation*. IEEE, 2019. 2

[25] Piotr Mirowski, Matthew Koichi Grimes, Mateusz Malinowski, Karl Moritz Hermann, Keith Anderson, Denis Teplyashin, Karen Simonyan, Koray Kavukcuoglu, Andrew Zisserman, and Raia Hadsell. Learning to navigate in cities without a map. *arXiv preprint arXiv:1804.00168*, 2018. 2

[26] Piotr Mirowski, Razvan Pascanu, Fabio Viola, Hubert Soyer, Andrew J Ballard, Andrea Banino, Misha Denil, Ross Goroshin, Laurent Sifre, Koray Kavukcuoglu, et al. Learning to navigate in complex environments. *arXiv preprint arXiv:1611.03673*, 2016. 2

[27] Matthias Müller et al. Driving policy transfer via modularity and abstraction. *arXiv preprint arXiv:1804.09364*, 2018. 2

[28] Emilio Parisotto and Ruslan Salakhutdinov. Neural map: Structured memory for deep reinforcement learning. *arXiv preprint arXiv:1702.08360*, 2017. 2

[29] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei Efros. Context encoders: Feature learning by inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[30] Aravind Rajeswaran, Sarvjeet Ghotra, Balaraman Ravindran, and Sergey Levine. Epopt: Learning robust neural network policies using model ensembles. *arXiv preprint arXiv:1610.01283*, 2016. 2

[31] Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal value function approximators. In *International Conference on Machine Learning*, pages 1312–1320, 2015. 2

[32] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017. 4, 5, 6

[33] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[34] Richard S Sutton, Joseph Modayil, Michael Delp, Thomas Degris, Patrick M Pilarski, Adam White, and Doina Precup. Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 761–768. International Foundation for Autonomous Agents and Multiagent Systems, 2011. 4

[35] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. MIT Press, 2005. 2

[36] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, pages 23–30. IEEE, 2017. 2

[37] Fei Xia, Amir R Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: Real-world perception for embodied agents. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 3, 5

[38] Amir R. Zamir, Alexander Sax, William B. Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018. 2, 3, 5, 6, 8

[39] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014. 2

[40] Fangyi Zhang, Jürgen Leitner, Michael Milford, and Peter Corke. Sim-to-real transfer of visuo-motor policies for reaching in clutter: Domain randomization and adaptation with modular networks. *world*, 7:8, 2017. 2

[41] 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 *IEEE International Conference on Robotics and Automation*, pages 3357–3364. IEEE, 2017. 1, 2, 3