FLAR: A Unified Prototype Framework for Few-sample Lifelong Active Recognition

Lei Fan, Peixi Xiong, Wei Wei and Ying Wu
Northwestern University, 2145 Sheridan Road, Evanston, IL, USA
{leifan, peixixiong2018, weiwei2022}@u.northwestern.edu, yingwu@northwestern.edu

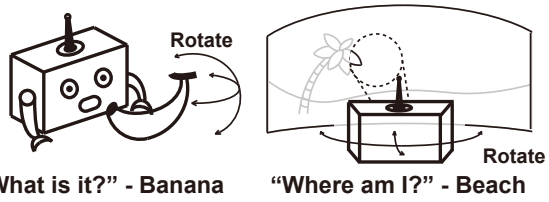
Abstract

Intelligent agents with visual sensors are allowed to actively explore their observations for better recognition performance. This task is referred to as Active Recognition (AR). Currently, most methods toward AR are implemented under a fixed-category setting, which constrains their applicability in realistic scenarios that need to incrementally learn new classes without retraining from scratch. Further, collecting massive data for novel categories is expensive. To address this demand, in this paper, we propose a unified framework towards Few-sample Lifelong Active Recognition (FLAR), which aims at performing active recognition on progressively arising novel categories that only have few training samples. Three difficulties emerge with FLAR: the lifelong recognition policy learning, the knowledge preservation of old categories, and the lack of training samples. To this end, our approach integrates prototypes, a robust representation for limited training samples, into a reinforcement learning solution, which motivates the agent to move towards views resulting in more discriminative features. Catastrophic forgetting during lifelong learning is then alleviated with knowledge distillation. Extensive experiments across two datasets, respectively for object and scene recognition, demonstrate that even without large training samples, the proposed approach could learn to actively recognize novel categories in a class-incremental behavior.

1. Introduction

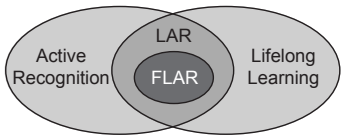
Visual recognition has been widely studied and achieved remarkable success in recent decades. In contrast to passive recognition from a still image, in robot learning scenarios, an intelligent agent is allowed to explore different viewpoints and is equipped with the capability to make decisions about what to observe. This problem is referred to as *Active Recognition (AR)*, with two specific tasks illustrated in Figure 1(a).

A number of *AR* methods [4, 18, 12, 6, 23, 13, 7] have been proposed over the years with learning-based models.



(a) The schematic illustrations of two *AR* tasks: active 3D object recognition and panoramic scene recognition. The system could intelligently select actions to acquire better views.

(b) A demonstration of lifelong learning during robot exploration. The system needs to expand its knowledge to novel categories that are discovered continuously.



(c) A diagram depicting the relation with other tasks. The terms *LAR* and *FLAR* are abbreviations for *Lifelong Active Recognition* and *Few-sample Lifelong Active Recognition*.

Figure 1. *FLAR* is one challenging task that requires expanding dynamically to novel classes with few training samples. Meanwhile, the task setting fits the practical needs of robotic applications that need exploring previously unseen environments.

Despite achieving promising results, these methods are confined to a classical learning setting, *i.e.*, the recognition decision can only be made for the samples from the trained categories, and massive training data are usually required to facilitate the learning process. When it comes to more practical settings where novel categories are continuously emerging, and more notoriously, only a few samples are available for the emerging category, it is not clear whether these models still remain effectual.

This problem is surprisingly under-explored in the literature but is imperative in realistic autonomous agent applications. In many scenarios, the agent trained in the backend to actively recognize fixed categories will be required to incrementally expand its *AR* ability for novel classes on the fly. We call this problem *Lifelong Active Recognition (LAR)*, illustrated in Figure 1(b). Further, it is often costly to collect many training samples for novel categories, let alone the possibility that the samples of the novel category are scarce *per se*. This motivates us to study a new problem, *Few-shot Lifelong Active Recognition (FLAR)*, that is both necessary and challenging. In Figure 1(c), the relationship of *FLAR* with closed problems is depicted.

Formally, *FLAR* raises three requirements to the agent. (1) The agent should be capable of making decisions to explore the most informative viewpoints based on the current stage so as to direct senses for a better understanding of surroundings. This fits into the realm of *AR* [32, 6, 4, 20, 41]. (2) The agent should adapt the power of exploration and recognition learned from old classes to new concepts while avoiding training from scratch. It is related to *incremental learning* [31, 30, 36]. (3) The agent should learn new concepts from limited training samples. It is connected with *few-shot learning* [37, 33, 15]. These requirements compose our *FLAR* problem, which delivers an intelligent agent that could incrementally learn to explore and recognize novel categories with only a few training samples.

Corresponding to these demands, three major challenges are posed by *FLAR*. (1) The previous AR methods typically learn a recognition policy from categories with massive training data. The constraint of few training samples for incremental categories will certainly impede the success of the policy training. (2) In the incremental learning setting, the agent is evaluated by the recognition performance of not only the categories on the fly but also old categories. Thus the catastrophic forgetting issue [30] needs to be tackled when new categories continuously emerge. (3) The risk of overfitting always exists for few-sample learning. Generalization from few samples needs to be fulfilled in our setting. In summary, *FLAR* is highly unconstrained with complex viewing conditions, growing recognition categories, and limited training samples.

In this paper, we propose a novel approach towards *FLAR*, a challenging but practical task that is under-explored. Although the challenges of *FLAR* scatter into different research fields, we address them in a unified framework. The main idea is that we hypothesize there exists a prototype in the embedding space to represent each category by averaging the aggregations of the budgeted exploratory observations for each sample. This facilitates flexible policy learning while simplifies knowledge preservation during class-incremental learning. Then for novel emerging categories, the agent is only required to take movements for the purpose of distinguishing the newly obtained features with prototypes of the trained categories. Note that the exemplars that best estimate prototypes are delicately selected and stored in the agent memory, which would be instilled during novel category learning.

Specifically, the insights of the proposed method aiming at the challenges of *FLAR* are three-fold: (1) The agent learns the active recognition policy based on a newly designed reward that favors a closer distance between aggregated features and the correct class prototype in the embedding space. (2) To handle the forgetting issue, only limited exemplars are stored in the agent memory in prioritized order. By reproducing consistent outputs for the exemplars utilizing the knowledge distillation mechanism, we incorporate the distribution of the old classes during learning novel concepts. (3) The prototype of each category, which serves as robust representations, potentially makes our approach adaptive to the few-training-sample challenge.

2. Related work

Active vision. Active vision has a long history in literature, which is pioneered by [2, 1, 8]. The common motivation behind these works is to bring intelligent control strategies to different visual tasks, *i.e.*, the agent should actively obtain observations with its own purpose. Following this idea, active vision has been exploited in several lines, covering tasks like recognition [34, 4, 18, 26, 24, 38], navigation [39, 11, 5], localization [3] and scene completion [19, 29].

As a notable branch of active vision, prior *AR* approaches can be mainly identified into two groups based on whether explicitly measuring information gains between different views. For the method describing gains explicitly, [32] proposes a 3D saliency model to guide action selections. Others [6, 4] perform information gain maximization by representing the problem as a partially observable Markov decision process (POMDP). These methods tend to disambiguate among candidate labels with view-specific profits. On the other hand, there are approaches that undertake *AR* with deep reinforcement learning methods, in which policies are learned by gathering interaction experiences with the environment. For example, in [18], three modules, including control, single view recognition and evidence fusion, are composed in an end-to-end trainable system. The auxiliary task of predicting future observations helps to build correlations between views and movements. [12] considers establishing a consistent 3D latent model for each category by estimating depth and ego-motions from images. The policy aggregates the latent map recurrently to predict actions. In stark contrast to the *FLAR* task, most existing *AR* methods are performed with predetermined categories and do not support expanding to new classes. However, the agent exploration is inherently incremental: novel classes

that could not be known in advance are demanded to be incorporated by the agent continuously.

Lifelong learning. Lifelong learning [27, 9, 30, 31, 36, 17, 35, 16, 21], also named continual learning, remains a long-standing challenge for machine learning since the catastrophic forgetting always occurs in non-stationary data distributions. Class-incremental learning requires progressively adding novel classes without restarting training from scratch. A training strategy towards class-incremental learning criteria is firstly proposed in [30], where the knowledge distillation is utilized to maintain information from previous time points. [31] pays attention to incremental learning on the few-shot image classification by introducing meta-learning on an attention module. Other current works [36, 16] also intend to implement continual learning under various difficult conditions. Our approach, on the other hand, focuses on *FLAR* which comprises lifelong learning on the sequence-based decision and recognition process.

Recent adaptive agents methods [39, 25, 28, 14] adopt continual learning techniques for dynamic environments. The robot movement policy proposed in [28] could evolve with kinematic changes, *e.g.*, to lose a wheel during exploration. These prospective works [39, 25] share similar motivation with our work, which is to address difficulties introduced by non-stationary environments. However, their method is unsuitable for *FLAR* since the knowledge preservation is not a precedent concern in these works.

Few-shot learning. There are plenty of works showing interest in few-shot image classification [33, 15, 37, 22, 42]. Their few-training-sample setting is closely related to the *FLAR* task. Various approaches have been delivered and could be roughly categorized into model-based, metric-based and optimization-based methods. In [33], they address few-shot image classification in the sense of metric learning by building a deep network as a function that could map the same-class inputs to the neighboring area in the embedding space. MAML [15] instead intends to learn a good parameter initialization of the base learner through gradients back-propagation. Our work shares a similar assumption with [33], *i.e.*, the classifier should have a simple inductive bias to prevent overfitting with few samples. Following this assumption, our work towards *FLAR* introduces prototype representation during policy learning, which prompts an effective policy to obtain more informative views.

3. Method

For ease of presentation, we first define the setup and notations for *FLAR*. Then we describe three significant components of the proposed method and explain how their combination allows performing *FLAR*. An overview of our approach is demonstrated in Figure 2.

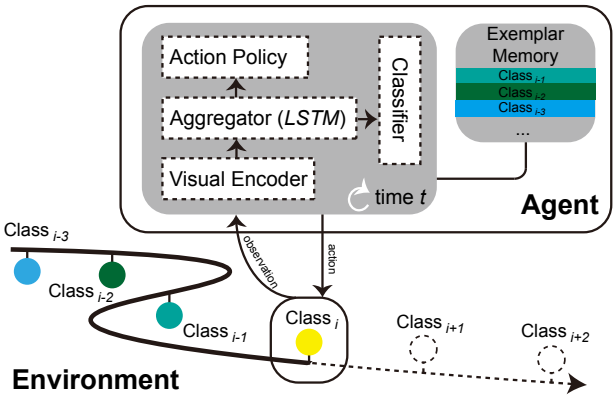


Figure 2. An overview of the proposed approach towards *FLAR*. Each task is denoted with a colored dot. The agent could interact with the environment by obtaining observations and making movements, which benefits recognition. As the agent exploring in the environment, the proposed method expands its recognition ability to new classes.

3.1. Problem setup and notation

We describe our setting by applying it to an active object recognition scenario.

The active object recognition $Agent$ is given an object instance x with an unknown label y. A total of T timesteps is allowed for the $Agent$ to predict the category of the object. At timestep $t = 1, 2, \ldots, T-1$, the $Agent$ could additionally select an action $a \in \mathcal{A}$, *e.g.*, to rotate up the object 30 degrees, where $\mathcal{A}$ denotes the action space. As a result of taking movement, the visual sensor mounted on the $Agent$ could get new observations of target instance x. We assume the visual sensor remains at the same position while only rotating the object. To be more specific, the visual observation at time t is a 2D view as $v_t = \mathcal{P}(x, p_t)$, where $\mathcal{P}(\cdot, \cdot)$ is a projection function and p_t is the corresponding viewpoint. We evenly discretize the space of all viewpoints into a view-grid with the size of M azimuths $\times$ N elevations. Then, each viewpoint can be designated to $p_t = (m, n)$ with $m \in M, n \in N$. The objective of $Agent$ during a recognition episode is, therefore, three-fold, including making efficient exploration, aggregating observations among timesteps, and classification based on the fused information.

After introducing the recognition setting, we then characterize the detailed setup of incremental learning. As the $Agent$ exploring in the environment, novel classes could occur at any time. A recognition task X indicates learning active recognition on specified categories. The $Agent$ exploration can then be described as a class-incremental task stream $X^{base}, X^1, X^2, \ldots, X^y, \ldots$. The X^{base} is the initial training set before the $Agent$ exploration, which is from C^{base} categories. Each following task, as training samples

of novel categories C^{novel}, is $X^y = \{x_1^y, \dots, x_k^y\}, y \in C^{novel}$. Since the high expense of collecting training samples for newly discovered categories $y \in C^{novel}$, we limit the samples in X^y by letting $|X^y| = k$, where k is limited to $3, 5, 10$ in our setting. For evaluation, the system is tested on the accuracy of its category prediction $\hat{y}$, and $\hat{y}$ belongs to seen candidate categories, *i.e.*, $\hat{y} \in C^{seen}$ with $C^{seen} = C^{base} \cup C^{novel}$.

3.2. Prototype-guided active recognition

We comprehend *AR* as a procedure of achieving more discriminative features by reaching different views. Let us recall the basic motivation of *AR*, which is based on an observation that a single static image might not include enough information for classification, especially in an unconstrained environment. In other words, a static image might not be discriminative enough. The action selection of the agent can then be seen as a policy on the feature space, which should be rewarded if its representation becomes easier to differentiate from candidate categories.

In this part, we introduce our *AR* system in four steps. Firstly, we describe the representations that we want to learn during policy training. Then we introduce our recognition system architecture to achieve objective representations. The novel reward designed on our representation, as motivating attaining better views, is described. Finally, other losses for training our *AR* system are provided.

Prototype representation learning With a 3D object instance x, our recognition system could obtain a 2D view projection at time t, which is denoted as $v_t = \mathcal{P}(x, p_t)$. The current view v_t together with other proprioceptions, including both the relative position $p_{t-1,t}$ and the timestep t itself, are regarded as the observations which is denoted as $\mathcal{X}_t = h(v_t, p_{t-1,t}, t)$, where $h(\cdot)$ is a fuse operation.

We present our representation module as $f \rightarrow \mathbb{R}^d$, a network to aggregate observations among timesteps. After actively taking $T-1$ movements, the representation for object x is $q_x = f(\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_T)$. Since training samples are limited, we hypothesize a prototype representation around which are aggregated features for the same class. Since the feature q_x is extracted from multi-views and their relative poses, the correlation between views and viewpoints could be obtained during training, which, in other words, describes the object shape. Compared to the few-shot image classification method [33], the restrictive description ability of only one prototype could be alleviated to some extent by absorbing shape information.

For each category y, the prototype comes from a collected set $Q^y = \{q_1^y, q_2^y, \dots\}$ as $\mu^y = \frac{1}{|Q^y|} \sum_{q \in Q^y} q$. Then, for an object instance x with the representation q_x, its label is assigned as:

$$\hat{y} = \mathbf{argmin}_y ||q_x - \mu^y||. \tag{1}$$

The label assignment is equal to $\hat{y} = \mathbf{argmax}_y \mu^y q_x$ after we normalize representations. Therefore, we could consider the prototype of each category as a weight vector from our final linear layer, which is multiplied with q_x for category probabilities. During training our *AR* system, we form the prototype representation learning with a loss term defined as:

$$\mathcal{L}_{category} = -\sum_i F_{softmax}(\hat{y}^i, y^i), \tag{2}$$

where $F_{softmax}$ is the softmax function, and the superscript i denotes the corresponding training sample.

Active recognition system Our *AR* system is modeled on the architecture proposed in [18], which is mainly composed of three modules. The first module performs as a non-linear mapping function, which is previously defined as $f(\cdot)$. In our approach, we utilize a combination of a visual encoder and an LSTM network to recurrently fuse observations. The second module, *i.e.*, the policy, could be treated as a Partially Observable Markov Decision Process (POMDP), whose pdf is defined as $\pi(a_t|\mathcal{X}_{t-1}, \theta)$. θ is the parameter we want to obtain with policy gradients. This module is represented as a combination of linear layers in our approach, which predicts action distributions with the aggregated features. The third module is the classifier, *i.e.*, a linear layer with the weight of prototypes for each category. At each timestep t, the proposed *AR* system selects an action with the highest probabilities. The classification is then applied to obtained temporally aggregated features.

Rewards for discrimination We design a novel reward to motivate the agent to choose views that result in more discriminative features. According to our classification in Equation 1, the discrimination ability between a feature and prototypes could be defined on their Euclidean distances. Intuitively, an increase of probabilities on the correct category represents the new feature becomes closer to the correct prototype among all candidates. We then define the reward $R(\hat{y}_t, \hat{y}_{t+1}) = 1$ as the growth of predicted probability of correct category, which promotes policy learning. Compared to a simple reward as $R(\hat{y})_t = 1$ when the category prediction is correct, our proposed reward always focuses on achieving better views progressively.

The reward is used to train the policy via the reinforcement learning technique, *i.e.*, REINFORCE, which could be back propagated to non-stochastic units. We define the loss for our policy learning as:

$$\mathcal{L}_{policy} = \sum_i \sum_{t=1}^{T-1} \log \pi(a_t^i|\mathcal{X}_{t-1}, \theta) R(\hat{y}_t, \hat{y}_{t+1})^i. \tag{3}$$

Other Losses Two other terms, *i.e.*, $\mathcal{L}_{entropy}$ and $\mathcal{L}_{forecast}$ are included during training our *AR* system. To promote more exploratory behavior of our agent and prevent policy collapse, the entropy loss $\mathcal{L}_{entropy}$ is calculated on the action distribution, which prefers selecting diversified actions.

Another term $\mathcal{L}_{forecast}$ shares the same idea with [18] by introducing an auxiliary task of forecasting observations. Formally, we define this term as the following:

$$\mathcal{L}_{forecast} = \sum_{i} \sum_{t=2}^{T} D(\hat{\mathcal{X}}_t^i, \mathcal{X}_t^i | \mathcal{X}_{t-1}^i, a_{t-1}), \quad (4)$$

where D is a similarity measure as the cosine distance.

3.3. Lifelong learning on novel classes

The proposed approach to this point could only perform *AR* on fixed categories. Its classifier could not accommodate new classes coming during exploration. In this part, we present further details of our approach in handling lifelong learning.

Agent memory Catastrophic forgetting happens in our setting. One way to handle this challenge is to entangle weight in the classifier with the representation learning process. If not, the final output would change out of management [30]. The weight in our classifier is set to prototypes that change along with our representations learning. Therefore, the data distribution of previous classes should be introduced to the current training process to track prototype changes. We build a memory to store exemplars, *i.e.*, object instances, that best describe the current category.

Only limited m exemplars would be stored for each category to save memory space. The exemplar selection is conducted in a prioritized fashion [30]. An exemplar is selected if, by adding it to the memory, the average feature vector would best approximate the prototype overall training data. Such selection processes can be done one time for each category. The saved exemplar set for category y is $M^y = \{x_1, x_2, \ldots, x_m\}$. In specific, the exemplars are saved in the form of view-grids for our approach since they are the direct visual inputs.

Distillation loss The knowledge of previous classes is maintained by encouraging reproducing the same output of saved exemplars. We implement recognition episodes on exemplars to achieve their classifier outputs z before training on new classes. The knowledge distillation mechanism is adopted as a loss term $\mathcal{L}_{distillation}$:

$$\mathcal{L}_{distillation} = -\sum_{i} \sum_{y \in C^{known}} F_{BCE}(z_i^y, f(x_i^y)), \quad (5)$$

where C^{known} is the category with exemplars at current, and F_{BCE} denotes the Binary Cross Entropy function.

To sum up, our approach to *FLAR* could be trained in an end-to-end trend with the following loss:

$$\begin{aligned} \mathcal{L} = \mathcal{L}_{category} + \mathcal{L}_{policy} + \mathcal{L}_{entropy} \\ + \mathcal{L}_{forecast} + \mathcal{L}_{distillation}. \end{aligned} \quad (6)$$

Each term is balanced with a constant that is ignored here. Note that the gradients of $\mathcal{L}_{policy}$ work only on the policy module, while other loss terms are effective on all modules except the policy module. We show the training procedure of the proposed approach in Algorithm 1 with one task from the task stream. After training finished, the class of our recognition system could be successfully extended.

Algorithm 1: Training on task X^i

Input: X^i: current task from the stream
Require: f: recurrent embedding module
Require: $Agent$: *AR* system with policy π
Require: $M^{i-1} = \{M^y, y \in C^{known}\}$: memory
$\mathcal{D} = X^i \cup M^{i-1}$: training set
// store network updates for the distillation loss
for $y \in C^{known}$ **do**
  Perform *AR* for all $x_i \in \mathcal{D}$ to get z_i^y
end
// network training
while *epoch reaches maximum* **do**
  Perform *AR* for all $x_i \in \mathcal{D}$
  Back propagate $\mathcal{L}$ defined in Equation 6
end
Update $C^{known} \leftarrow C^{known} \cup y^i$
Update agent memory to M^i

4. Experiments

To validate our approach to *FLAR*, we examine the performance on two challenging datasets. We first introduce the utilized datasets and our experimental setups. Then, we evaluate our approach in the class-incremental setting in Section 4.3. The comparison of our approach with other baselines is demonstrated in Section 4.4, which shows the effectiveness of our policy in *AR*. In Section 4.5, we conduct ablation studies on the training sample size and the saved exemplar size.

4.1. Datasets and experimental setups

We evaluate our approach on two widely used datasets for scene and object recognition, respectively.

SUN360 dataset The SUN360 scene dataset [40] contains spherical panoramas for 26 diverse scene categories. The dataset is split into 6174 training, 1013 validation, and 1805 testing examples. We test our agent on this dataset for active scene recognition. The field-of-view of our agent is limited to 60 degrees. The agent could rotate to move to novel observations (shown in Figure 3). The agent needs an efficient policy to obtain good scene recognition accuracy in limit steps which is set to $T = 5$. We discretize the panorama into a grid with 32 views, *i.e.*, elevations $M = 4$ and azimuths $N = 8$, which is the same setting in [29]. Each view is a 32 pixels $\times$32 pixels 2D image. We set the action space of our agent as a 3×5 view-grid centered at the current position. In other words, the agent movement is restricted to a

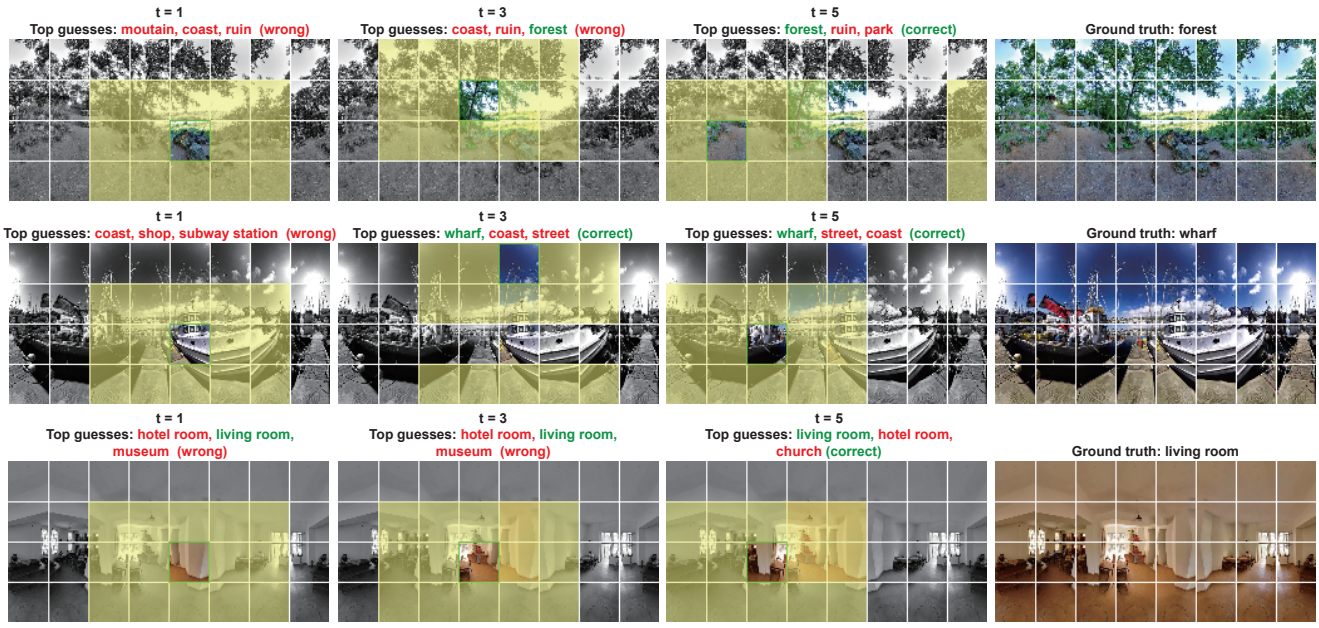


Figure 3. The active scene recognition process of the proposed approach. Each row contains 3 steps, *i.e.*, $t = 1, 3, 5$, from a recognition episode. The starting position is set to the same to show different trajectories on three samples. We mark the current view (green box) and the next movement grid (light yellow area). As shown in the first row, the proposed approach corrects its reasonable but wrong guesses within 5 steps. More visualizations of results from the ShapeNet dataset are included in the supplementary material.

3×5 grid at each timestep.

We arrange the SUN360 dataset into a task stream to suit *FLAR*. We randomly select 16 categories as the initial task with all training samples, which forms the base category C^{base}. After the initial task, the agent is trained in a class-incremental way on the following 10 categories with limited samples. Each category represents a novel task. The performance is evaluated on test samples of the dataset, considering all classes that have already been trained. The amount k of new-category training samples is limited, which is set to 5 if not specified. Moreover, only $m = 3$ exemplars are saved to memory without designation.

ShapeNet datasets Our experiment conducted on the ShapeNet dataset [10] considers the scenario that the agent could manipulate an object instance for recognition. The agent needs to predict its next best motion based on previous observations. Each training sample is a Computer-Aided Design (CAD) model. The view with the resolution of 32×32 is sampled from $M = 6$ camera elevations and $N = 12$ azimuths. For each step, the agent is able to move within 5 elevations-by-7 azimuths neighborhood of the current position. A total of 5 views could be achieved before giving final category predictions.

We randomly select 20 object categories from the ShapeNet dataset to conduct our experiments. Each category contains 35 samples for training, 10 samples for validation, and 10 samples for evaluation. Among 20 categories, we select 10 categories as the base category and form the other 10 categories into sequential tasks. The ShapeNet dataset is more challenging than the SUN360 scene dataset for two reasons. Firstly, it contains more view grids than in our SUN360 dataset setting, which, in other words, brings larger searching space for policy learning. Secondly, the synthetic 3D model might contain less texture information than a real object. And we also evaluate the performance of our approach with all seen categories.

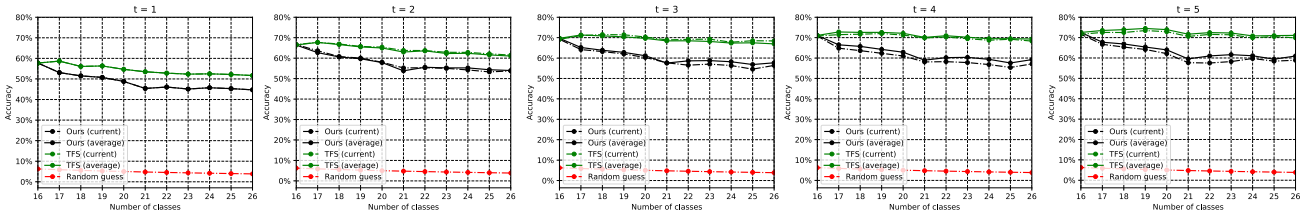
4.2. Implementations

Our approach is implemented with PyTorch. The visual encoder of our approach is a simple 3-layer network with the ReLU activation. We utilize the recurrent neural network (LSTM) to aggregate temporal knowledge from observations. During trajectory gathering in reinforcement learning, we randomly provide the starting viewpoints. The exemplars are saved to the memory during training. Currently, we have not considered the limit of total memory size, which would be considered in our future work. We attach a classifier, *i.e.*, a linear layer without bias, to the LSTM output at each step. The final classification result is reported as the average of class likelihoods of reached steps. We use `current` to denote the result based only on the current estimates and `average` as the final result.

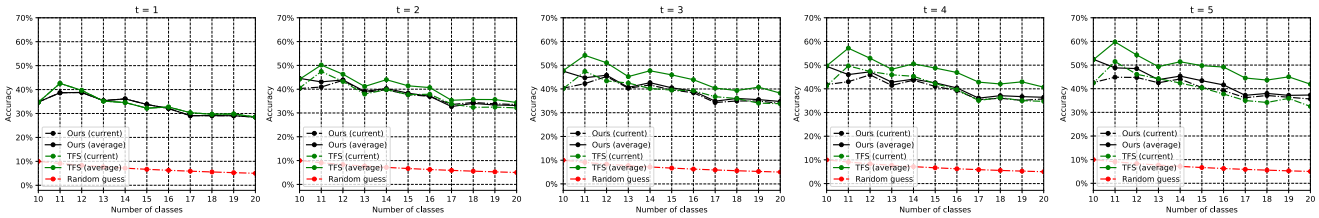
4.3. Lifelong learning results

In this part of the experiment, we study the performance of the proposed approach under the full *FLAR* condition.

Continual learning In this part, we intend to demonstrate the effectiveness of our approach in dealing with learning



(a) The result of the proposed approach on the SUN360 dataset [40].



(b) The result of the proposed approach on the ShapeNet dataset [10].

Figure 4. Recognition accuracy for both datasets. The method `TFS` is short for Training From Scratch, which could access sufficient data of all categories. The method `Random Guess` defines the lower bound of our performance.

novel categories. Since the proposed approach, to the best of our knowledge, is the first to address *FLAR*, we attempt to define the range of our performance by dramatically ease the task. We introduce the training setting that we name it `Train From Scratch` (`TFS`). The `TFS` could access all training data at the same time, which, in other words, is not constrained by both the forgetting or the few-sample challenges. Therefore, the closer of our performance with the result of `TFS` denotes the higher effectiveness of the proposed method, as we could achieve similar results by progressively learning novel classes with few samples.

Figure 4 shows the results on two datasets. The metric been utilized is classification accuracy. Note the `TFS` is re-trained with all data on each category setting while the proposed approach incrementally learns on novel classes without access to previous data. For each dataset, we display the result with timestep $t = 1, 2, \ldots, 5$. The performance of the proposed approach is the same as the `TFS` on the base category since no class-incremental learning is performed. One can see the effectiveness of our method in learning novel classes, especially for the ShapeNet dataset [10]. The advantage comes from two parts. Firstly, the reward during our policy learning motivates the agent to take actions towards differentiating with other known categories. Secondly, the concept of previous classes could be maintained by the knowledge distillation on the agent exemplar memory.

As expected, the overall performance of the proposed approach is better on the SUN360 dataset than on the ShapeNet dataset since the searching space for the ShapeNet dataset is obviously larger. Another finding revealed in Figure 4 is the performance arises with taking more steps. We will show the evaluation of our policy in Section 4.4 to explain the improvement is not only brought by obtaining more observations but also the policy.

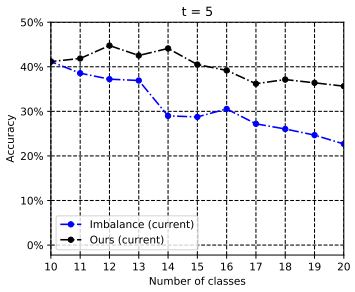


Figure 5. Evolution of accuracy v.s. the number of classes for different learning strategies. Our results show improvements over the result from `Imbalance` setting that is confronted with the data-imbalance problem.

An interesting observation from Figure 4 is the proposed approach occasionally exceeds the performance of the `TFS` on the ShapeNet dataset [10]. It first positively suggests the prototype representation is effective in handling few-sample challenges. It also indicates large samples of a category might not bring direct benefits to the prototype representation since the prototype, *i.e.*, the mean of features, would be distracted by several "hard" training samples.

Imbalance of training data We show the advantage of class-incremental learning from the aspect of data balance. We combine both samples from base categories and novel categories. Compared to `TFS` that can have sufficient training samples for novel categories, only limited k samples are provided in the `Imbalance` setting. Direct training under the `Imbalance` setting could be regarded as a long-tail/data-imbalance problem. In Figure 5, we demonstrate the `Imbalance` result with ours with incremental learning strategy. Our results at $t = 5$ steadily outperform the results from the `Imbalance` setting, which shows the effectiveness of our incremental learning strategy.

Method	t = 2 acc.		t = 3 acc.		t = 5 acc.	
	curr.	*avg.*	*curr.*	*avg.*	*curr.*	*avg.*
Single view	51.6	51.6	51.6	51.6	51.6	51.6
Random views	55.6	56.5	57.7	59.1	59.8	62.3
Largest step	54.7	55.7	53.6	56.6	52.4	55.8
Look-Ahead [18]	59.8	60.2	67.8	66.3	69.4	70.6
Ours	**61.5**	**61.0**	**68.4**	**67.0**	**69.9**	**71.1**

Table 1. Recognition accuracy on the SUN360 dataset [40]. The *curr.* denotes results with current estimates while *avg.* is the average of class likelihoods up to the current step.

4.4. Comparison on *AR*

In this part, we would like to show the effectiveness of our policy on *AR* solely. We block our mechanism on class-incremental learning leaving only an *AR* agent for fixed categories. We first introduce the baselines.

`Single view`: The input is only a random starting view to our approach. No policy is needed in this method. We include this method in our comparison to show the performance of single view recognition.

`Random views`: This method shares a similar architecture with the proposed method, which replaces our policy module with a random action selection. The number of movements remains the same as ours.

`Largest step`: The policy is to take the movement that is the most distant to the current viewpoint. The assumption here is neighbor views usually share similar information.

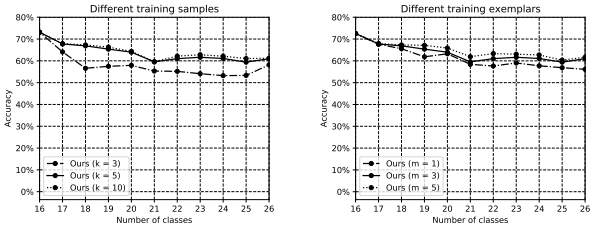
`Look-Ahead`: This method [18] is also based on a recurrent network architecture. The reward is defined as the current movement getting the correct prediction, which is different from our approach. It runs to the fixed steps as ours.

The comparisons on the SUN360 dataset [40] with all 26 categories are displayed in Table 1. The proposed approach could already outperform other methods on the basic *AR* task. The large improvements between our results with other passive baselines, including `Single view`, `Random views` and `Largest step`, denote the advantages of including effective policies during recognition. Our method is also better than [18] which is the result of two attributes. The first is our prototype representation learning which promotes obtaining structure consistencies between different object instances. The second reason is our novel reward that always motivates the agent to achieve more informative views. Note that the agent could intelligently stop taking further movements by redefining the action space.

4.5. Ablation studies

To provide further details of our approach, in this part, we perform our methods on the SUN360 dataset [40], in which we isolate its individual aspects.

Sample size First, we analyze the influence of the sample size on our method. We set the sample size $k = 3, 5, 10$ while keeping the other parameters the same. The number of exemplars saved to memory is $m = 3$. Our method is then trained in these three different settings. Figure 6(a) summarizes the results as the accuracy over all steps of the class-incremental learning. The results show that the number of samples actually contributes to the performance. In particular, by comparing the results of $k = 5$ and $k = 10$, one can see that the performance growth is not significant. We think it suggests that the proposed approach could obtain adequate prototype representations with 5 training samples. Another observation is the performance goes up after learning on class 26. The reason could be that our network achieves better prototypes of previous categories with saved exemplars during training on class 26, or class 26 introduces beneficial transferable knowledge.



(a) Training sample sizes (b) Exemplar sizes

Figure 6. Ablation studies on the SUN360 dataset [40].

Exemplar size We study how the number of exemplars influences our performances. The exemplar is chosen in a prioritized order after training on the current category. The result in Figure 6(b) are trained with $m = 1, 3, 5$ with $k = 5$. Note $m = 1$ means only one exemplar that most approximate to the prototype is stored in the agent memory. The result demonstrates the effectiveness of the proposed approach when the memory size is extremely limited, which, in other words, validates our exemplar selection process.

5. Conclusions

In this paper, we propose a novel approach towards *FLAR* which incrementally learns active recognition on novel categories. Challenges, including few training samples and forgetting, are addressed with three major components. We derive the prototype as the representation for each category, which is robust in handling limited training samples. The novel designed reward motivates the agent to achieve more discriminative features by measuring distances in the embedding space. To alleviate catastrophic forgetting, the knowledge distillation with exemplars stored in the agent memory is applied during the policy training. The experimental results, along with ablation studies, show the effectiveness of proposed approach for the *FLAR* task. However, despite the promising result, *FLAR* is still a challenging task only at the beginning stage. We plan to study the influence of category relations to *AR* in our future work.

Acknowledgements

This work was supported in part by National Science Foundation grant IIS-1619078, IIS-1815561, and IIS-2007613.

References

[1] John Aloimonos. Purposive and qualitative active vision. In *Proceedings of the International Conference on Pattern Recognition*, 1990. 2

[2] John Aloimonos, Isaac Weiss, and Amit Bandyopadhyay. Active vision. *International Journal of Computer Vision*, 1988. 2

[3] Alexander Andreopoulos and John K Tsotsos. A theory of active object localization. In *IEEE International Conference on Computer Vision*, 2009. 2

[4] Alexander Andreopoulos and John K Tsotsos. A computational learning theory of active object recognition under uncertainty. *International Journal of Computer Vision*, 2013. 1, 2

[5] Nikolay Atanasov, Jerome Le Ny, Kostas Daniilidis, and George J Pappas. Decentralized active information acquisition: Theory and application to multi-robot slam. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4775–4782. IEEE, 2015. 2

[6] Nikolay Atanasov, Bharath Sankaran, Jerome Le Ny, George J Pappas, and Kostas Daniilidis. Nonmyopic view planning for active object classification and pose estimation. *IEEE Transactions on Robotics*, 2014. 1, 2

[7] Ruzena Bajcsy, Yiannis Aloimonos, and John K Tsotsos. Revisiting active perception. *Autonomous Robots*, 42(2):177–196, 2018. 1

[8] Dana H Ballard. Animate vision. *Artificial intelligence*, 1991. 2

[9] Abhijit Bendale and Terrance Boult. Towards open world recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1893–1902, 2015. 3

[10] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 6, 7

[11] Devendra Singh Chaplot, Dhiraj Gandhi, Abhinav Gupta, and Ruslan Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. *arXiv preprint arXiv:2007.00643*, 2020. 2

[12] Ricson Cheng, Ziyan Wang, and Katerina Fragkiadaki. Geometry-aware recurrent neural networks for active visual recognition. *arXiv preprint arXiv:1811.01292*, 2018. 1, 2

[13] Joachim Denzler and Christopher M Brown. Information theoretic sensor data selection for active object recognition and state estimation. *IEEE Transactions on pattern analysis and machine intelligence*, 24(2):145–157, 2002. 1

[14] Yan Duan, John Schulman, Xi Chen, Peter L Bartlett, Ilya Sutskever, and Pieter Abbeel. Rl ²: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*, 2016. 3

[15] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, 2017. 2, 3

[16] Jiangpeng He, Runyu Mao, Zeman Shao, and Fengqing Zhu. Incremental learning in online scenario. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13926–13935, 2020. 3

[17] Khurram Javed and Martha White. Meta-learning representations for continual learning. *arXiv preprint arXiv:1905.12588*, 2019. 3

[18] Dinesh Jayaraman and Kristen Grauman. Look-ahead before you leap: end-to-end active recognition by forecasting the effect of motion. In *European Conference on Computer Vision*, 2016. 1, 2, 4, 5, 8

[19] Dinesh Jayaraman and Kristen Grauman. Learning to look around: Intelligently exploring unseen environments for unknown tasks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2

[20] Edward Johns, Stefan Leutenegger, and Andrew J Davison. Pairwise decomposition of image sequences for active multi-view recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3813–3822, 2016. 2

[21] Jean Kaddour, Steindór Sæmundsson, and Marc Peter Deisenroth. Probabilistic active meta-learning. *arXiv preprint arXiv:2007.08949*, 2020. 3

[22] Bingyi Kang, Zhuang Liu, Xin Wang, Fisher Yu, Jiashi Feng, and Trevor Darrell. Few-shot object detection via feature reweighting. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019. 3

[23] S Kasaei, Juil Sock, Luis Seabra Lopes, Ana Maria Tomé, and Tae-Kyun Kim. Perceiving, learning, and recognizing 3d objects: An approach to cognitive service robots. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018. 1

[24] Sena Kiciroglu, Helge Rhodin, Sudipta N Sinha, Mathieu Salzmann, and Pascal Fua. Activemocap: Optimized viewpoint selection for active human motion capture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 103–112, 2020. 2

[25] Vincenzo Lomonaco, Karan Desai, Eugenio Culurciello, and Davide Maltoni. Continual reinforcement learning in 3d non-stationary environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2020. 3

[26] Mohsen Malmir, Karan Sikka, Deborah Forster, Ian Fasel, Javier R Movellan, and Garrison W Cottrell. Deep active object recognition by joint label and action prediction. *Computer Vision and Image Understanding*, 156:128–137, 2017. 2

[27] Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriela Csurka. Metric learning for large scale image classification: Generalizing to new classes at near-zero cost. In *European Conference on Computer Vision*, 2012. 3

[28] Anusha Nagabandi, Ignasi Clavera, Simin Liu, Ronald S Fearing, Pieter Abbeel, Sergey Levine, and Chelsea Finn. Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. *arXiv preprint arXiv:1803.11347*, 2018. 3

[29] Santhosh K Ramakrishnan, Dinesh Jayaraman, and Kristen Grauman. Emergence of exploratory look-around behaviors

through active observation completion. *Science Robotics*, 2019. 2, 5

[30] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2, 3, 5

[31] Mengye Ren, Renjie Liao, Ethan Fetaya, and Richard S Zemel. Incremental few-shot learning with attention attractor networks. *arXiv preprint arXiv:1810.07218*, 2018. 2, 3

[32] Andrea Roberti, Marco Carletti, Francesco Setti, Umberto Castellani, Paolo Fiorini, and Marco Cristani. Recognition self-awareness for active object recognition on depth images. In *BMVC*, 2018. 2

[33] Jake Snell, Kevin Swersky, and Richard S Zemel. Prototypical networks for few-shot learning. *arXiv preprint arXiv:1703.05175*, 2017. 2, 3, 4

[34] Stefano Soatto. Actionable information in vision. In *Machine Learning for Computer Vision*. 2013. 2

[35] Stefan Stojanov, Samarth Mishra, Ngoc Anh Thai, Nikhil Dhanda, Ahmad Humayun, Chen Yu, Linda B Smith, and James M Rehg. Incremental object learning from contiguous views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8777–8786, 2019. 3

[36] Xiaoyu Tao, Xiaopeng Hong, Xinyuan Chang, Songlin Dong, Xing Wei, and Yihong Gong. Few-shot class-incremental learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2, 3

[37] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. *arXiv preprint arXiv:1606.04080*, 2016. 2, 3

[38] Wei Wei, Haonan Yu, Haichao Zhang, Wei Xu, and Ying Wu. Metaview: Few-shot active object recognition. *arXiv preprint arXiv:2103.04242*, 2021. 2

[39] Mitchell Wortsman, Kiana Ehsani, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Learning to learn how to learn: Self-adaptive visual navigation using meta-learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2, 3

[40] Jianxiong Xiao, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Recognizing scene viewpoint using panoramic place representation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 5, 7, 8

[41] Jianwei Yang, Zhile Ren, Mingze Xu, Xinlei Chen, David J Crandall, Devi Parikh, and Dhruv Batra. Embodied amodal recognition: Learning to move to perceive objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2040–2050, 2019. 2

[42] Xiangyun Zhao, Yi Yang, Feng Zhou, Xiao Tan, Yuchen Yuan, Yingze Bao, and Ying Wu. Recognizing part attributes with insufficient data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 350–360, 2019. 3