NeRS: Network-Based Reinforced System for Embodied Intelligence

Junhua Liu
The Chinese University of Hong Kong,Shenzhen

Abstract—**Inspired by the risen of generalist dataset Gato and first-person perception dataset Ego4d, which firstly works as a multi-modal, multi-task, multi-embodiment generalist policy, the visual-language tasks have accelerated into a new era. There has been an emerging paradigm shift from the "interest AI" to "embodied AI" in vision-language tasks. The implementation of embodied AI also encounter several problems. Except for the optimization of better performance and new dataset, the implementation of the embodied AI is also decisive.**

In this paper, we propose NeRS, a network-based reinforced system for embodied AI, which mainly focused on: (1) Learn how visual representations pre-trained on dataset can enable data-efficient learning of downstream robotic manipulation tasks. (2) Propose a robustness and low-cost network system for transmission under dynamic bandwidth, bitrate and rebuffer. (3) Enhance the generalization of embodied AI tasks under long-term and dynamic environment that outperform existing visual-language solutions. (4)Explore the feasibility of combining machine learning system and deep learning and design a tiny machine learning pipeline to assembly the model trained online.NeRS is the first model to be pre-trained on Gato and Ego4d. Our evaluations shows the generalization on Long trajectories and multiple input types, and the robust network outperform existing video streaming by 37%. Project page and source code is available at https://github.com/JunhuaLiu0/NeRS.

Index Terms—**Embodied AI, Multimodal System, Computer Vision, 360° video streaming, Online Machine Learning, Tiny Machine Learning, Computer Network.**

I. Introduction

Recent advances in deep learning, reinforcement learning, computer graphics and robotics have garnered growing interest in developing general-purpose AI systems. As a result, there has been a shift from "internet AI" that focuses on learning from datasets of images, videos and text curated from the internet, towards "embodied AI" which enables artificial agents to learn through interactions with their surrounding environments. Embodied AI is the belief that true intelligence can emerge from the interactions of an agent with its environment [1], which is about incorporating traditional intelligence concepts from vision, language and reasoning into an artificial embodiment to help solve AI problem in a virtual environment, With the integration and flourish of embedded system, it perceives and acts within real environments in the future [2].

In Embodied AI, the task can be classifed into three types: Visual Language Navigation(VLN), Visual Language Exploration(VLE) and first-person video understanding [1].This task needs to understand both the natural language instructions and the visible image information in the visual Angle, and then make corresponding actions to its own state in the environment, and finally reach the target position. In VLE, the agent did VLE to learn from the environment. In VLN, the intelligent agent follows the natural language instructions to navigate. In first person video understanding, the robot is asked to do specific movement in the first person viewport.

Gato, works as a multi-modal, multi-task, multi-embodiment generalist policy [3]. There are significant benefits to using a single neural sequence model across all tasks. It reduces the need for hand crafting policy models with appropriate inductive biases for each domain. It increases the amount and diversity of training data since the sequence model can ingest any data that can be serialized into a flat sequence. Furthermore, its performance continues to improve even at the frontier of data, compute and model scale [4]. Ego4D is the largest dataset of first-person views to date, with a total of 3,600 + hours of video footage, each cut into clip-level clips, each with a one-sentence text description. The risen of dataset Gato and Ego4D accelerate the process of implementation. We can pre-train in the new situation and fine-tuning the new model, which will be used in downstream tasks but cost less.

With new paradigm of *Metaverse*, the magnitude of Embodied AI is large and synthesis. Despite the potentials, the implementation of Embodied AI faces a key challenge of generalization. Embodied AI is also exigent and high-cost. This came to a head recently with the release of the GPT-3 algorithm [5], boasting a network architecture containing a staggering 175 billion neurons, which cost a lot to train, used numerous electricity. The implementation of Embodied AI is also dependent on high-quality graphics processing units(GPU), which can be very expensive. So we propose a reinforced network system to transmit the panoramic video and do the online end-to-end learning on cloud server. The server receive the training data(panoramic video) from network, send the strategy and the training model.

For vision-language task, the computation cost of training neural network is very large, especially with the increase of resolution, the computation increases at least linearly. The nature of human vision is not to process the whole scene at once, but to process the image sequence, obtain the key position of the image, extract key location information and then combine to construct the whole scene information. The process can be concluded as Attention mechanism. Soft attention is widely used in recent computer vision tasks. But

Junhua Liu is with the School of Data Science, the Chinese University of Hong Kong, Shenzhen, China, 518172. (e-mail: junhualiu@link.cuhk.edu.cn)

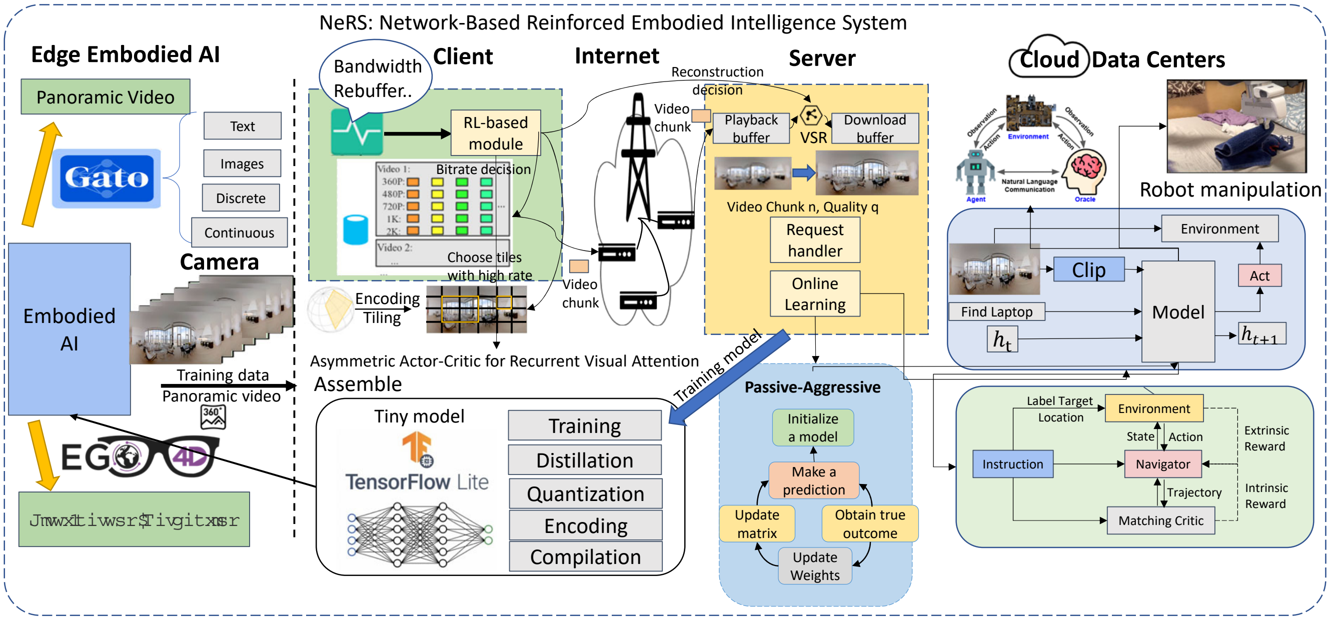


Figure 1. System Design of NeRS

it is slower to train and inference according to preliminary experiment, and the generalization is also unpromising. For hard attention (Recurrent Visual Attention), it uses an anchor of fixed size, the extraction is based reinforcement learning, which is quickly to inference. But it is hard to converge when training in complex data because of discrete value 0 and 1, and also have the delayed reward.

To solve the hardness of converging, we use Asymmetric Actor-Critic for Recurrent Visual Attention as reference and combine the feature of panoramic video. The panoramic video can be divided into several tiles of same size. The anchor is the combination of one or several tiles, which makes it easier to converge. .

Machine learning (ML) inference on the edge is an increasingly attractive prospect due to its potential for increasing energy efficiency [6], privacy, responsiveness [7], and autonomy of edge devices. A specific example is TinyML. The goal of “TinyML” is to bring ML inference to ultra-low-power devices. The efficiency of TinyML enables a class of smart, battery-powered, always-on applications. That can revolutionize the real-time collection and processing of data [8]. TinyML can plays an irreplaceable role in the large-scale implementation of embodied AI because of its fitness in embedded system and low cost.

In this paper, we mainly propose NeRS, a network-based reinforced system for embodied intelligence, which donate a solution on the implementation of Embodied AI. The tasks of Embodied AI can be divided into two categories. Our system is the first one to use Gago and Ego4D as pre-trained dataset. We use reinforcement learning(RL) as the backbone of the system in the face of generalization and assists in network system. Our framework uses internal and external rewards for reinforcement learning, and introduces a circular reconstruction reward as an internal reward to force agents to globally match instructions and trajectory. There is a big difference in the effect between the seen scene and the unseen scene. We combine RL with imitation learning(IL) to boost the generalization. The system propose an online end-to-end learning method, use Passive-aggressive, an online machine learning algorithm to update the weight and matrix, and use Clip, a method that connection text and image to cross-modally match the instruction and navigation. We Design the pipeline of TinyML, including Training, distillation, quantization, encoding and compilation. In navigation on real world, the network bandwidth and rebuffer can be dynamic and unstable. The panoramic video can be equirectangularly projected in 2D surface, which can be divided into several tiles. Our system propose a mechanism learned from Asymmetric Actor-Critic for Recurrent Visual Attention. We use reinforcement learning to control the resolution of different block, the resolution of high attention tile will be enhanced, and the resolution of low attention tile will be reduced. The quality of overall video will be adjusted according the network condition.

Overall, the design of system include the implementation of Embodied AI. The main contributions of our paper are as follows:

- NeRS is the first model to be pre-trained on Gato and Ego4d. We Learn the result of pre-trained and fine-tuning on large-scale dataset and how it performs on downstream tasks like manipulation and visual-language tasks.
- Propose a robustness and low-cost network system for transmission under dynamic bandwidth, bitrate and rebuffer, which is based on reinforcement learning.
- Combining reinforcement learning and imitation learning to enhance the generalization of embodied AI tasks under long-term and dynamic environment, which outperform existing visual-language solutions.
- Uses internal and external rewards for reinforcement learning, and introduces a circular reconstruction reward

as an internal reward to force agents to globally match instructions and trajectory. Design a tiny machine learning pipeline to assembly the model trained online.

II. Related Work

A. Embodied AI

*1) **Vision-Language Exploration and Navigation**:* Recent research in feature extraction [9]–[13]. attention [10], [14], [15] and multi-modal grounding [16], [17] have helped the agent to understand the environment. Researchers in both computer vision and natural language processing are striving to bridge vision and natural language towards a deeper understanding of the world. Previous works in Vision-Language Navigation have focused on improving the ability of perceiving the vision and language inputs [18]–[20] and cross-modal matching [21], [22].

It is difficult to reason about visual images and natural language instructions, and the agent needs to match the parts of the visual scene and instructions in the path. However, clip [23] outperforms the existing solutions on the connectivity of image and text. Moreover,However, those tasks focus on passive visual perception in the sense that the visual inputs are usually fixed. In this work, we are particularly interested in solving the dynamic multi-modal grounding problem in both temporal and spatial spaces. Thus, we focus on the task of vision-language navigation (VLN), which requires the agent to actively interact with the environment. There is a big generalization problem between the effect of seen scenes and unseen scenes.

*2) **First-person perception**:* Prior works have explored the use of natural language in robot manipulation, primarily as a means of task specification or reward learning [24]. Current studies on first-person video comprehension tasks can be divided into two categories based on the sources of training sets. The first category is pretraining directly using non-first-person video data sets such as Kinetics. The advantage of this is that the video source is rich and there is a large amount of data for training.

The difficulty is how to transfer the knowledge from non-first-person video to first-person video task. [25] The second more common method is to directly use first-person video data sets for training. However, due to the high cost of such video collection, previous data sets are limited by data scale and single scene [26]. On the contrary, we use diverse human video data and language annotations to learn reusable visual representations for control. Prior work has also found visual representations informed by language, Through empirical evaluations.

B. Reinforcement learning

1) Overview: Reinforcement Learning [27] is appropriate to adaptively optimize the long-term reward. RL seeks to numerically conclude the states of a given system, and execute an action to optimize an accumulated reward. Deep learning further enables RL to scale to decisionmaking problems that were previously intractable, i.e., the problems with settings of high-dimensional state space and action space [27]. Many researchers have proposed RL-based models to learn the relationships between the features and the objectives, and achieved success in various application. The agent feedback success only after reaching the target instruction, but ignores whether to follow the path of the instruction.

2) Asymmetric Actor Critic: Asymmetric actor critic had been utilized in the field of robotics [28] to help close the bridge between simulation and the real world. As robots are slow and expensive and thus data intensive learning algorithms are hard to scale, physical robotics could potentially be damaged during actual learning process. A common approach is to circumvent with training in a simulated system of the robot and then transfer the skill under real setting. This approach, though solves the problem of scalability, brings the new challenge of observability. Under the real learning process, the robots often have limited observations and thus the policy learnt during simulated progress can be difficult to deploy. The asymmetric actor critic method provides a mean to allow robots learn a policy based of partial observations in the real world while keeping the policy trained on full state information during the simulated learning process being useful.

3) Recurrent Visual Attention: Sample inefficiency have always been a great interest of study in the area of reinforcement learning. In recent years, functional approximation with neural networks have received more and more interests. In 2017, the Proximal Policy Optimization (PPO) [29] was proposed with the vision to improve the current state of art reinforcement learning algorithms with only first order optimization. The classical policy gradient method was modified to optimize a "surrogate" objective function, which have been defined in three different ways in the original paper. Experiments included in this paper claims that PPO outperforms other state of the arts algorithms such as Actor-Critic.

C. Video Transmission and Streaming

The viewport of Embodied AI can be regarded as a 360° video(panoramic video). 360° video is a new kind of video representation that is recorded by omnidirectional cameras and can provide an immersive and interactive watching experience to viewers. The video content forms as a sphere rather than a plane, where the viewer can freely adjust his/her head angle to watch a portion of content in the sphere. But in the stage of coding, the sphere is actually projected to a 2D plane using such projection methods as equirectangular or cube-map projections because existing encoders work on 2D rectangles.

Given such spherical features, streaming 360° videos requires much higher network bandwidth than traditional videos. To accommodate the high resource consumption, tile-based video encoding/streaming together with FoV-adaptive tile selection is proposed. Each frame of a video is split into different tiles and only tiles inside the user's FoV are streamed at high quality [30]–[34]. Petrangeli et al. [35] show an HTTP/2-based adaptive streaming framework to achieve higher performance. Nasrabadi et al. [36] propose a Scalable Video Coding encoding method so that the number of video rebuffering events can be reduced. Zare et al. [37] use the motion-constrained tile set feature of High Efficiency Video Coding standard to tackle the problem of multiple decoders at the user side to decode each tile. Guan et al. [38] leverage the 360° video-specific factors and Dasari et al. [39] use super Super-Resolution to save the bandwidth.

D. Tiny ML

Previously, complex circuitry was necessary for a device to perform a wide range of actions. Now, machine learning is making it increasingly possible to abstract such hardware into software, making embedded devices increasingly simple, lightweight, and flexible. Great progress has already been achieved in this area. The key challenges in deploying neural networks on microcontrollers are the low memory footprint, limited power, and limited computation [6], [7], [40], [41]. TinyML within smartphones to detect the specific words like ”Hey Siri”. Specialized low-power hardware that is able to be powered by a small battery [6], [7], [40]. Keyword spotting: devices listen continuously to audio input from a microphone and are trained to only respond to specific sequences of sounds, which correspond with the learned keywords [7], [42].Visual wake words: an image-based analog to the wake words known as visual wake words [7], [40], [43].

III. Network System

A. RSS: RL-based Streaming with Super-Resolution

The streaming of panoramic video can be formatted as the problem of bitrate adaptation with super-resolution. **Video streaming with super-resolution**. In RSS, we introduce the super-resolution to the adaptive video streaming with the reconstruction decision. Regarding the i-th chunk of the video, a reconstruction decision can be expressed by (q_i, q_i) indicating that the chunk downloaded with the q_i (named as the base resolution) resolution would be reconstructed to the q i resolution (named as the target resolution). The server should reconstruct the low-resolution video chunk based on the reconstruction decision in real-time. The monitor collects the potential improvement of the VSR and the reconstruction time of the previous video chunks into the playback statistics. Based on the playback statistics, the RL-based bitrate adaptation module could determine whether it is necessary to download a low-resolution chunk and reconstruct it to high resolution. Hence, the appropriate super resolution model in SSR should not be too complex, or the super-resolution would never be called due to the minimal rebuffering concern.

Agent-environment Interface: In the RL semantic, the decision-maker is called the agent. The agent keeps interact with the environment for a specific objective. At each step, the agent makes an action according to its state. The environment responds to the action with a numerical reward and updates the state of the agent. The goal of the RL procedure is to maximize the rewards over steps under a policy which guides the agent to generate actions. In RSS, client can be regarded as the agent in the playback environment. The action is to make the reconstruction decision over the states during the playback, aiming to maximize the numerical QoE objective.

The Training of RL-based streaming will shown in Evaluation(A).

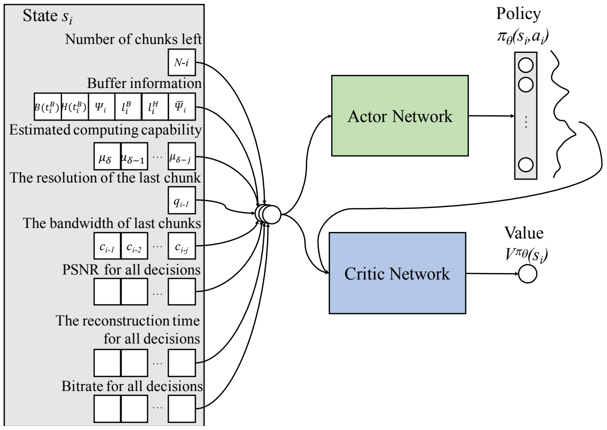


Figure 2. The architecture of the RL-based module based on A3C structure.

B. BAAA: Bitrate Allocation With Asymmetric Actor-Critic

For the navigation in real world, the network bandwidth and rebuffer can be dynamic and unstable when suffering from weak signal. The online end-to-end training is a continuous process. The panoramic video can be equirectangularly projected in 2D surface, which can be divided into several tiles. The resolution of each tile can be adaptively allocated. Our system propose a mechanism learned from Asymmetric Actor-Critic for Recurrent Visual Attention. We train a RL-based model to control the resolution of different block, the resolution of high attention tile will be enhanced, and the resolution of low attention tile will be reduced. The quality of overall video will be adjusted according to the network condition by RSS.

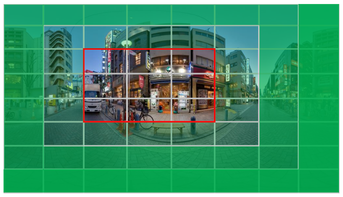


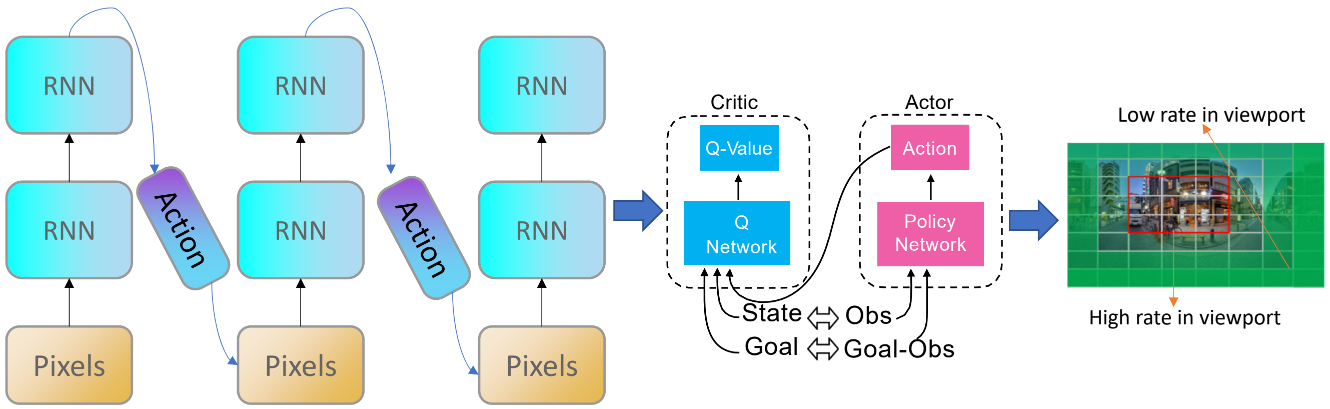
Figure 3. Tile-based streaming in panoramic video

For hard attention (Recurrent Visual Attention), it uses an anchor of fixed size, the extraction is RL-based, which is quickly to inference. But it is hard to train in the face of complex data because the value is discrete and can only be 0 or 1, and also have the delayed reward, which leads to the misconvergence. In Recurrent Visual Attention, the train dataset is about the single digital number and the anchor is a window of fixed size. The model can be simiplied when focusing on the control of tiles. With reinforcement learning, the anchor can change adaptively, it can be the combination of one or several tiles, which makes the process easier to converge.

IV. Reinforced Embodied AI

A. Overview

Here we consider an embodied agent that learns to navigate inside real indoor environments by following natural language



instructions. The RCM framework mainly consists of two modules: a reasoning navigator and a matching critic V. Given the initial state s_0 and the natural language instruction (a sequence of words) X = $x_1, x_2, ..., x_n$, the reasoning navigator learns to perform a sequence of actions $a_1, a_2, ..., a_T$ A, which generates a trajectory , in order to arrive at the target location starget indicated by the instruction X . The navigator interacts with the environment and perceives new visual states as it executes actions. To promote the generalizability and reinforce the policy learning, we introduce two reward functions: an extrinsic reward that is provided by the environment and measures the success signal and the navigation error of each action, and an intrinsic reward that comes from our matching critic and measures the alignment between the language instruction X and the navigator's trajectory

B. CLIP in Embodied AI

In the previous section, we showed that CLIP representations are remarkably effective across multiple embodied tasks, especially compared to their counterparts. We wish to bring some insight as to why this is the case. To this end, we design a collection of experiments measuring how well visual representations from ImageNet and CLIP pretrained models encode the following semantic and geometric primitives: object presence, object presence at a location, object reachability, and free space. We posit that visual representations that encode these primitives effectively will lead to more capable Embodied AI agents, particularly given the nature of the 4 tasks we are investigating. For each of these primitives, we train simple (linear) classifiers to predict the outcome of interest from image features

C. Model

Online End-to-End Learning model can be divided by:

1) Navigator: The navigator is a policy-based agent that maps the input instruction X onto a sequence of actions $at_{t=1}^{T}$. At each time step t, the navigator receives a state s_t from the environment and needs to ground the textual instruction in the local visual scene. Thus, we design a cross-modal reasoning navigator that learns the trajectory history, the focus of the textual instruction, and the local visual attention in order, which forms a cross-modal reasoning path to encourage the local dynamics of both modalities at step t.

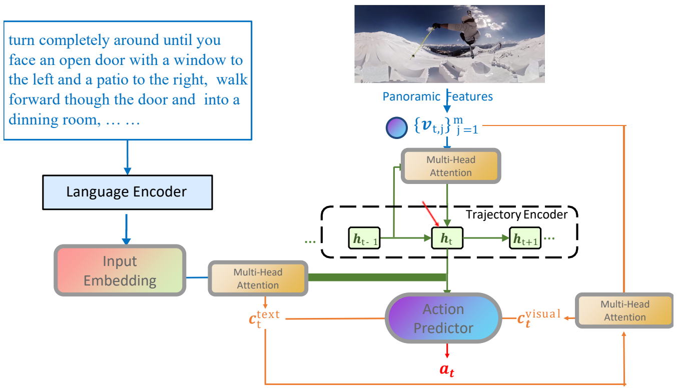


Figure 4. Navigator and Critic

Figure shows the unrolled version of the navigator at time step t. We equip the navigator with a panoramic view, which is split into image patches of m different viewpoints, so the panoramic features that are extracted from the visual state st can be represented as $v_{t,j_{mj}} = 1$, where $v_{t,j}$ denotes the pre-trained CNN feature of the image patch at viewpoint j.

History Context Once the navigator runs one step, the visual scene would change accordingly. The history of the trajectory $\tau 1 : t$ till step t is encoded as a history context vector ht by an attention-based trajectory encoder LSTM [44]:

$$h_t = LSTM$$

([vt, at1], ht1) (1)

where $a_t 1$ is the action taken at previous step, and $v_t = \sum_j \alpha_{t,j} v_{t,j}$, the weighted sum of the panoramic features. $\alpha_{t,j}$ is the attention weight of the visual feature $v_{t,j}$, representing its importance with respect to the previous history context h_{t1}. Note that we adopt the dot-product attention here after, which we denote as (taking the attention over visual features above for an example)

$$
\begin{aligned}
v_t = & \text{ attention } \left(h_{t-1}, \{v_{t,j}\}_{j=1}^{m}\right) \\
= & \sum_j \operatorname{softmax}\left(h_{t-1} W_h \left(v_{t,j} W_v\right)^T\right) v_{t,j}
\end{aligned}
$$

where W_h and W_v are learnable projection matrices.

Visually Conditioned Textual Context Memorizing the past can enable the recognition of the current status and thus understanding which words or sub-instructions to focus on next. Hence, we further learn the textual context ctext t conditioned on the history context ht. We let a language encoder LSTM to encode the language instruction X into a set of textual features wini=1. Then at every time step, the textual context is computed as

$$c_t^{\text{text}} = \text{attention}\left(h_t, \{w_i\}_{i=1}^{n}\right) \tag{2}$$

Note that ctext t weighs more on the words that are more relevant to the trajectory history and the current visual state.

Textually Conditioned Visual Context Knowing where to look at requires a dynamic understanding of the language instruction; so we compute the visual context cvisual t based on the textual context ctext t:

$$c_t^{\text{visual}} = \text{attention}\left(c_t^{\text{text}}, \{v_j\}_{j=1}^{m}\right) \tag{3}$$

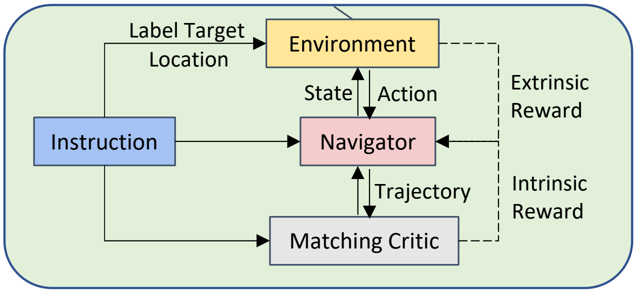


Figure 5. Navigator and Critic

Action Prediction In the end, our action predictor considers the history context ht, the textual context ctext t , and the visual context cvisual t , and decides which direction to go next based on them. It calculates the probability pk of each navigable direction using a bilinear dot product as follows:

$$p_k = \operatorname{softmax}\left(\left[h_t, c_t^{\text{text}}, c_t^{visual}\right] W_c \left(u_k W_u\right)^T\right) \tag{4}$$

where u_k is the action embedding that represents the kth navigable direction, which is obtained by concatenating an appearance feature vector (CNN feature vector extracted from the image patch around that view angle or direction) and a 4-dimensional orientation feature vector [sin; cos; sin; cos], where and are the heading and elevation angles respectively. The learning objectives for training the navigator are introduced in Section 3.3.

2) Critic: In addition to the extrinsic reward signal from the environment, we also derive an intrinsic reward Rintr provided by the matching critic V to encourage the global matching between the language instruction X and the navigator 's trajectory τ

One way to realize this goal is to measure the cyclereconstruction reward p(ˆX = X —(X)), the probability of reconstructing the language instruction X given the trajectory = (X) executed by the navigator. The higher the probability is, the better the produced trajectory is aligned with the instruction. Therefore as shown in Figure, we adopt an attentionbased sequence-to-sequence language model as our matching critic V, which encodes the trajectory with a trajectory encoder and produces the probability distributions of generating each word of the instruction X with a language decoder. Hence the intrinsic reward

$$R_{\text{intr}} = p_\beta\left(\mathcal{X} \mid \pi_\theta(\mathcal{X})\right) = p_\beta(\mathcal{X} \mid \tau) \tag{5}$$

which is normalized by the instruction length n. In our experiments, the matching critic is pre-trained with human demonstrations (the ground-truth instruction-trajectory pairs ¡ X , ¿) via supervised learning.

3) Imitative Reinforcement Learning: To learn a better and more generalizable policy, we then switch to reinforcement learning and introduce the extrinsic and intrinsic reward functions to refine the policy from different perspectives.

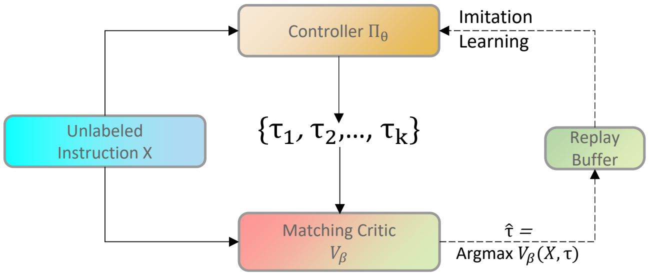


Figure 6. Controller and Critic

Extrinsic Reward A common practice in RL is to directly optimize the evaluation metrics. Since the objective of the VLN task is to successfully reach the target location starget, we consider two metrics for the reward design. The first metric is the relative navigation distance similar to [48]. We denote the distance between st and starget as Dtarget(st). Then the immediate reward r(st, at) after taking action at at state st (t ¡ T) becomes:

$$r\left(s_t, a_t\right) = \mathcal{D}_{\text{target}}\left(s_t\right) - \mathcal{D}_{\text{target}}\left(s_{t+1}\right), \quad t < T \tag{6}$$

This indicates the reduced distance to the target location after taking action at. Our second choice considers the "Success" as an additional criterion. If the agent reaches a point within a threshold measured by the distance d from the target (d is preset as 3m in the R2R dataset), then it is counted as "Success". Particularly, the immediate reward function at last step T is defined as

$$r\left(s_T, a_T\right) = \nVdash\left(\mathcal{D}_{\text{target}}\left(s_T\right) \leq d\right) \tag{7}$$

where () is an indicator function. To incorporate the influence of the action at on the future and account for the local greedy

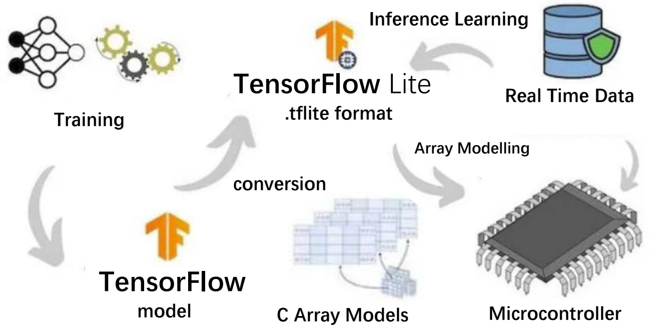


Figure 7. TinyML workflow

search, we use the discounted cumulative reward rather than the immediate reward to train the policy:

$$R_{\text{extr}}\ (s_t, a_t) = \underbrace{r(s_t, a_t)}_{\text{immediate reward}} + \underbrace{\sum_{t'=t+1}^{T} \gamma^{t'-t} r(s_{t'}, a_{t'})}_{\text{discounted future reward}}$$

where γ is the discounted factor (0.95 in our experiments).

Intrinsic Reward As discussed in Section 3.2.2, we pretrain a matching critic to calculate the cycle-reconstruction intrinsic reward Rintr (see Equation 8), promoting the alignment between the language instruction X and the trajectory . It encourages the agent to respect the instruction and penalizes the paths that deviate from what the instruction indicates. With both the extrinsic and intrinsic reward functions, the RL loss can be written as

$$L_{rl} = -\mathbb{E}_{a_t \sim \pi_\theta}[A_t]$$

where the advantage function $A_t = R_{\text{extr}} + \delta R_{\text{intr}}$.δ is a hyperparameter weighing the intrinsic reward. Based on REINFORCE algorithm [49], the gradient of nondifferentiable, reward-based loss function can be derived as

$$\nabla_0 L_{rl} = -A_t \nabla_0 \log \pi_0 (a_t \mid s_t)$$

4) Assembly in Embodied AI: Model Distillation:Post-training, the model is then altered in such a way as to create a model with a more compact representation. The process can be divided into Pruning and knowledge distillation.

The idea underlying **knowledge distillation** is that larger networks(teacher network) have some sparsity or redundancy within them. While large networks have a high representational capacity. if the network capacity is not saturated it could be represented in a smaller network(student network) with a lower representation capacity. The embedded information in the teacher model is transferred into the student model It is used to enshrine the same knowledge in a smaller network, providing a way of compressing networks such that they can be used on more memory-constrained devices.

Pruning can help to make the model's representation more compact. Pruning attempts to remove neurons that provide little utility to the output prediction, which is often associated with small neural weights, whereas larger weights are kept due to their greater importance during inference. The network is then retrained on the pruned architecture to fine-tune the output.

Quantization Following distillation, the model is then quantized post-training into a format that is compatible with the architecture of the embedded device. By quantizing the model, the storage size of weights is reduced by a factor of 4 (for a quantization from 32-bit to 8-bit values), and the accuracy is often negligibly impacted.

Huffman Encoding Encoding is an optional step that is sometimes taken to further reduce the model size by storing the data in a maximally efficient way: often via the famed Huffman encoding.

Compilation Once the model has been quantized and encoded, it is converted to a format that can be interpreted by some form of light neural network interpreter, the most popular of which are probably TF Lite (500 KB in size) and TF Lite Micro (20 KB in size). The model is then compiled into C or C++ code (the languages most microcontrollers work in for efficient memory usage) and run by the interpreter on-device.

V. Evaluation and Analysis

A. Network System

1) RSS: RL for Streaming with Super-resolution: **States**: In the streaming system, we collect some important information as states to provide evidence for clients to make decisions. The number of remaining chunks (N i) and the number of reconstructed chunks are important information in the playback environment. The bandwidth is the key limitation for decisions, so the average bandwidth when downloading the last few chunks ci1, ci2, ..., cik1 is used as an implicit prediction for the bandwidth, where k1 is a hyperparameter. Apart from the bandwidth condition, B(tB i) and H (tB i) are included to provide buffer condition. Let i denote the sum of standard reconstruction time of video chunks in downloading buffer, i provide information for prediction of the reconstruction time:

$$\Psi_i = \sum_{j=i-B(t_i^B)}^{i-1} \psi_j'(q_j, q_j') \tag{8}$$

We also take the elapsed time and standard reconstruction time of the video chunk being reconstructed, denoted as l_i^B, reconstruction time $\bar{\Psi}_i$, and the remaining time of the video chunk being played l_i^H into consideration:

$$l_i^B = \begin{cases} 0, & i - B(t_i^B) = \delta + 1 \\ t_i^B - t_{\delta+1}^H, & \text{otherwise} \end{cases}$$

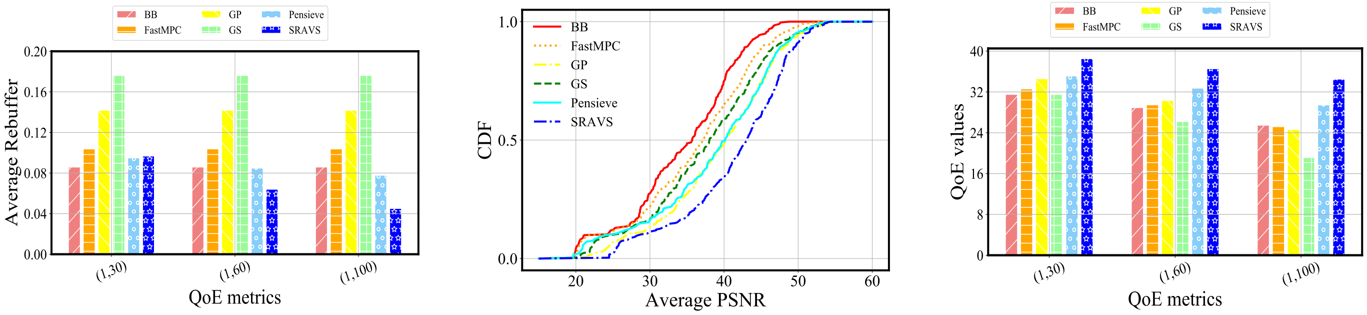
$$\bar{\Psi}_i = \begin{cases} 0, & i - B(t_i^B) = \delta + 1 \\ \bar{\psi}_{\delta+1}(q_{\delta+1}, q_{\delta+1}'), & \text{otherwise} \end{cases}$$

and

$$l_i^H = \max\left\{T - \left(t_i^B - t_{\delta - H(t_i^B)}\right), 0\right\}$$

Action: The RL-based model seeks to make reconstruction decision according to current state s_i, which is treated as the action $a_i = (q_i, q_i') \in \mathcal{D}$.

Reward: The monitor can record the rebuffering time τ_i during playback. Note that $\text{PSNR}(i, q_i, q_i')$ and $f(q_i')$ are fixed

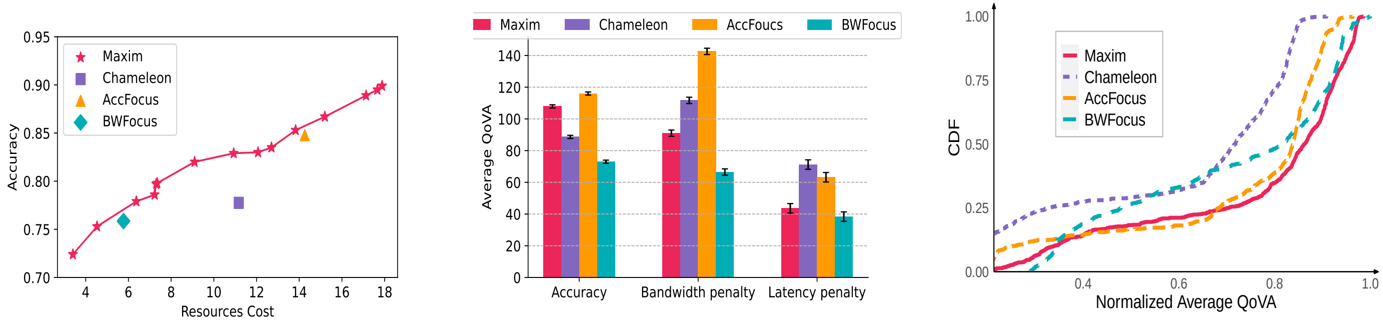


(a) The rebuffering by different streaming systems under various QoE objectives.

(b) The distribution of average PSNR of the compared streaming systems under QoE with

(c) The comparison of performance on QoE by different streaming systems.

Figure 8. Result of RSS



(a) Comparing Maxim with base-lines on invidividual terms in QoVA

(b) Comparing Maxim with base-lines on normalized average QoVA

(c) Evaluate Maxim and baselines on accuracy-resource trade-off

Figure 9. Result of BAAA

values for the certain reconstruction decision $(q_i, q_i') \in \mathcal{D}$. Hence, we can obtain the value of the QoE objectives in realtime and take it as the reward for the RL model, denoted as r_i for the i-th chunk, which would be calculated by:

$$r_i = \text{PSNR}\left(i, q_i, q_i'\right) - \alpha_1 \left| f\left(q_i'\right) - f\left(q_{i-1}'\right) \right| - \alpha_2 \tau_i$$

Policy: The goal of the reinforcement learning is to maximize the expected cumulative discounted reward, i.e., $\mathbb{E}\left[\sum_{i=1}^{N} r_i \gamma^i\right]$, where $\gamma \in (0, 1]$ is the discount rate of future rewards. The actions should be selected based on a stochastic policy, defined as a probability distribution over actions, denoted as $\pi_\theta\left(s_i\right) \rightarrow [0, 1]$, where θ represents the parameters of the RL model. $\pi_\theta\left(s_i\right)$ is the probability that action a_i is taken given the current state s_i. Then, Problem 1 can be transformed into:

$$\pi_\theta^* = \arg\max_{\pi_\theta} V^{\pi_\theta}\left(s_0\right),$$

where

$$V^{\pi_\theta}\left(s_i\right) = \mathbb{E}\left[\sum_{j=i}^{N} r_j \gamma^{j-i}\right]$$

is the state value function with policy $\pi_\theta(\cdot)$. The state value function measures the expectation of the cumulative discounted QoE reward by $\pi_\theta(\cdot)$ from the i-th chunk to the end of the playback.

Baseline methods. We compare the performance of RSS with the following video streaming system to verify that RSS does integrate the vision super-resolution(VSR) to the video streaming. • Bandwidth-based DASH (BB), which simply takes the average download speed of the last chunk as the prediction of bandwidth. BB seeks to maximize video quality without delay. The content in the buffer is not taken into consideration.

• FastMPC seeks to maximize the QoE by control theory to minimize the risk of suffering from poor QoE. FastMPC does not consider the super-resolution to video streaming.

• Greedy sequential (GS). GS takes the reconstruction process into consideration while the downloading and reconstruction must be executed in order. The bandwidth and reconstruction time are both predicted by linear regression (LR) in this streaming system.

• Greedy parallel (GP). GP takes the reconstruction process into consideration with both downloading buffer and playback buffer. The bandwidth and reconstruction time are both predicted by LR. GP uses a greedy strategy to make the reconstruction decision based on the maximal PSNR of the current chunk.

• Pensieve [9]. Pensieve leverages A3C architecture to learn a streaming strategy without super-resolution. Metrics. We select 3 sets of weights of (1, 2) to indicate 3 QoE objectives, namely (1,30), (1,60), (1,100), for various tolerance for latency.

B. Embodied AI tasks

]

VI. Conclusion

In this paper we present two novel approaches about network and reinforcement learning. We explore the dataset Gato and first-person perception dataset Ego4d, which firstly works as a multi-modal, multi-task, multi-embodiment generalist policy, the visual-language tasks have accelerated into a new era. We Learn how visual representations pre-trained on dataset

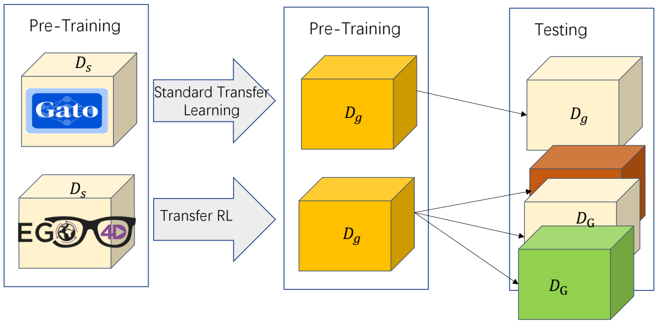


Figure 10. Pre-trained and fine-tuning

can enable data-efficient learning of downstream robotic manipulation tasks.Propose a robustness and low-cost network system for transmission under dynamic bandwidth, bitrate and rebuffer. Enhance the generalization of embodied AI tasks under long-term and dynamic environment that outperform existing visual-language solutions. Explore the feasibility of combining machine learning system and deep learning and design a tiny machine learning pipeline to assembly the model trained online.NeRS is the first model to be pre-trained on Gato and Ego4d. Our evaluations shows the generalization on Long trajectories and multiple input types, and the robust network outperform existing video streaming by 37%.

Acknowledgments

References

[1] J. Duan, S. Yu, H. L. Tan, H. Zhu, and C. Tan, "A survey of embodied AI: From simulators to research tasks." [Online]. Available: http://arxiv.org/abs/2103.04918

[2] Y. Lyu, Y. Shi, and X. Zhang, "Improving target-driven visual navigation with attention on 3d spatial relationships," *Neural Processing Letters*, pp. 1–20, 2022.

[3] S. Reed, K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-Maron, M. Gimenez, Y. Sulsky, J. Kay, J. T. Springenberg, T. Eccles, J. Bruce, A. Razavi, A. Edwards, N. Heess, Y. Chen, R. Hadsell, O. Vinyals, M. Bordbar, and N. de Freitas, "A generalist agent," number: arXiv:2205.06175. [Online]. Available: http://arxiv.org/abs/2205.06175

[4] J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. d. L. Casas, L. A. Hendricks, J. Welbl, A. Clark *et al.*, "Training compute-optimal large language models," *arXiv preprint arXiv:2203.15556*, 2022.

[5] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[6] I. Fedorov, R. P. Adams, M. Mattina, and P. Whatmough, "Sparse: Sparse architecture search for cnns on resource-constrained microcontrollers," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[7] Y. Zhang, N. Suda, L. Lai, and V. Chandra, "Hello edge: Keyword spotting on microcontrollers," number: arXiv:1711.07128. [Online]. Available: http://arxiv.org/abs/1711.07128

[8] C. R. Banbury, V. J. Reddi, M. Lam, W. Fu, A. Fazel, J. Holleman, X. Huang, R. Hurtado, D. Kanter, A. Lokhmotov, D. Patterson, D. Pau, J.-s. Seo, J. Sieracki, U. Thakker, M. Verhelst, and P. Yadav, "Benchmarking TinyML systems: Challenges and direction," number: arXiv:2003.04821. [Online]. Available: http://arxiv.org/abs/2003.04821

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[10] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems*, vol. 26, 2013.

[11] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, "Bottom-up and top-down attention for image captioning and visual question answering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6077–6086.

[12] R. Zeng, H. Xu, W. Huang, P. Chen, M. Tan, and C. Gan, "Dense regression network for video grounding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 287–10 296.

[13] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[15] J. Lu, J. Yang, D. Batra, and D. Parikh, "Hierarchical question-image co-attention for visual question answering," *Advances in neural information processing systems*, vol. 29, 2016.

[16] J. Lu, D. Batra, D. Parikh, and S. Lee, "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks," *Advances in neural information processing systems*, vol. 32, 2019.

[17] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, "Vqa: Visual question answering," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2425–2433.

[18] S. Gupta, J. Davidson, S. Levine, R. Sukthankar, and J. Malik, "Cognitive mapping and planning for visual navigation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2616–2625.

[19] D. Fried, R. Hu, V. Cirik, A. Rohrbach, J. Andreas, L.-P. Morency, T. Berg-Kirkpatrick, K. Saenko, D. Klein, and T. Darrell, "Speaker-follower models for vision-and-language navigation," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[20] X. Wang, W. Xiong, H. Wang, and W. Y. Wang, "Look before you leap: Bridging model-free and model-based reinforcement learning for planned-ahead vision-and-language navigation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 37–53.

[21] Y. Zhu, F. Zhu, Z. Zhan, B. Lin, J. Jiao, X. Chang, and X. Liang, "Vision-dialog navigation by exploring cross-modal memory," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 730–10 739.

[22] X. Wang, Q. Huang, A. Celikyilmaz, J. Gao, D. Shen, Y.-F. Wang, W. Y. Wang, and L. Zhang, "Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6629–6638.

[23] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning*. PMLR, 2021, pp. 8748–8763.

[24] S. Nair, E. Mitchell, K. Chen, S. Savarese, C. Finn *et al.*, "Learning language-conditioned robot behavior from offline data and crowd-sourced annotation," in *Conference on Robot Learning*. PMLR, 2022, pp. 1303–1315.

[25] Y. Li, T. Nagarajan, B. Xiong, and K. Grauman, "Ego-exo: Transferring visual representations from third-person to first-person videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 6943–6953.

[26] G. A. Sigurdsson, A. Gupta, C. Schmid, A. Farhadi, and K. Alahari, "Charades-ego: A large-scale dataset of paired third and first person videos," *arXiv preprint arXiv:1804.09626*, 2018.

[27] R. S. Sutton and A. G. Barto, "Reinforcement learning: an introduction mit press," *Cambridge, MA*, vol. 22447, 1998.

[28] L. Pinto, M. Andrychowicz, P. Welinder, W. Zaremba, and P. Abbeel, "Asymmetric actor critic for image-based robot learning," *arXiv preprint arXiv:1710.06542*, 2017.

[29] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[30] M. Graf, C. Timmerer, and C. Müller, "Towards bandwidth efficient adaptive streaming of omnidirectional video over HTTP: design, implementation, and evaluation," in *Proceedings of the 8th ACM on Multimedia Systems Conference, MMSys 2017, Taipei, Taiwan, June 20-23, 2017*. ACM, 2017, pp. 261–271.

[31] M. Hosseini and V. Swaminathan, "Adaptive 360 VR video streaming: Divide and conquer," in *Proceedings of the IEEE International Symposium on Multimedia, ISM 2016, San Jose, CA, USA, December 11-13, 2016*. IEEE Computer Society, 2016, pp. 107–110.

[32] P. Rondao-Alface, J. Macq, and N. Verzijp, "Interactive omnidirectional video delivery: A bandwidth-effective approach," *Bell Labs Tech. J.*, vol. 16, no. 4, pp. 135–147, 2012.

[33] X. Corbillon, G. Simon, A. Devlic, and J. Chakareski, "Viewport-adaptive navigable 360-degree video delivery," in *Proceedings of the IEEE International Conference on Communications, ICC 2017, Paris, France, May 21-25, 2017*. IEEE, 2017, pp. 1–7.

[34] V. R. Gaddam, M. Riegler, R. Eg, C. Griwodz, and P. Halvorsen, "Tiling in interactive panoramic video: Approaches and evaluation," *IEEE Trans. Multim.*, vol. 18, no. 9, pp. 1819–1831, 2016.

[35] S. Petrangeli, V. Swaminathan, M. Hosseini, and F. D. Turck, "An http/2-based adaptive streaming framework for 360° virtual reality videos," in *Proceedings of the 2017 ACM on Multimedia Conference, MM 2017, Mountain View, CA, USA, October 23-27, 2017*. ACM, 2017, pp. 306–314.

[36] A. T. Nasrabadi, A. Mahzari, J. D. Beshay, and R. Prakash, "Adaptive 360-degree video streaming using scalable video coding," in *Proceedings of the 2017 ACM on Multimedia Conference, MM 2017, Mountain View, CA, USA, October 23-27, 2017*. ACM, 2017, pp. 1689–1697.

[37] A. Zare, A. Aminlou, M. M. Hannuksela, and M. Gabbouj, "Hevc-compliant tile-based streaming of panoramic video for virtual reality applications," in *Proceedings of the 2016 ACM Conference on Multimedia Conference, MM 2016, Amsterdam, The Netherlands, October 15-19, 2016*. ACM, 2016, pp. 601–605.

[38] Y. Guan, C. Zheng, X. Zhang, Z. Guo, and J. Jiang, "Pano: optimizing 360° video streaming with a better understanding of quality perception," in *Proceedings of the ACM Special Interest Group on Data Communication, SIGCOMM 2019, Beijing, China, August 19-23, 2019*. ACM, 2019, pp. 394–407.

[39] M. Dasari, A. Bhattacharya, S. Vargas, P. Sahu, A. Balasubramanian, and S. R. Das, "Streaming 360-degree videos using super-resolution," in *Proceedings of the 39th IEEE Conference on Computer Communications, INFOCOM 2020, Toronto, ON, Canada, July 6-9, 2020*. IEEE, 2020, pp. 1977–1986.

[40] H. Ren, D. Anicic, and T. A. Runkler, "Tinyol: Tinyml with online-learning on microcontrollers," in *2021 International Joint Conference on Neural Networks (IJCNN)*, 2021, pp. 1–8.

[41] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *CoRR*, vol. abs/1804.02767, 2018.

[42] Y. Zhang, N. Suda, L. Lai, and V. Chandra, "Hello edge: Keyword spotting on microcontrollers," *CoRR*, vol. abs/1711.07128, 2017. [Online]. Available: http://arxiv.org/abs/1711.07128

[43] A. Chowdhery, P. Warden, J. Shlens, A. Howard, and R. Rhodes, "Visual wake words dataset," *CoRR*, vol. abs/1906.05721, 2019. [Online]. Available: http://arxiv.org/abs/1906.05721

[44] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

Junhua Liu is currently pursuing an B.S. degree at The Chinese University of Hong Kong, Shenzhen, China. His research interests include Multimedia Computing, Virtual Reality Video, and 3D Vision.