LOVON: Legged Open-Vocabulary Object Navigator

Author Names Omitted for Anonymous Review

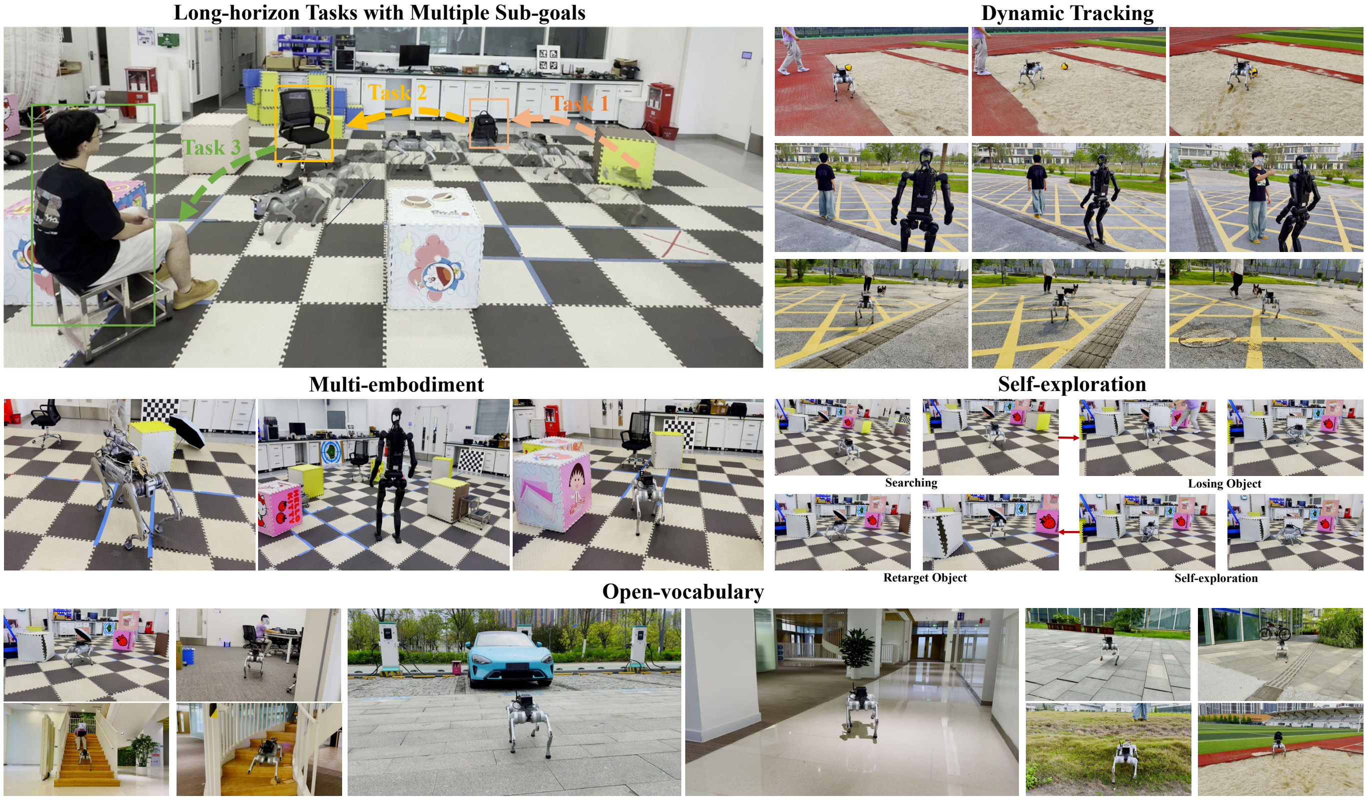

Fig. 1: **Legged robot object navigation in diverse open-world scenarios.**

***Abstract*—Object navigation in open-world environments remain a critical challenge for robotic systems. Despite advancements in large language models (LLMs) for task planning, open-vocabulary vision models for object detection, and versatile legged robots capable of traversing complex terrains, existing approaches lack a unified navigation framework to execute composite long-range missions. We propose LOVON, a novel system that integrates LLMs for hierarchical task planning with open-vocabulary visual detection and legged robot mobility. To address real-world challenges including visual jittering, blind zones, and temporary target loss, we design dedicated solutions such as Laplacian Variance Filtering for visual stabilization. Extensive evaluations on Go2, B2, and H1-2 legged platforms demonstrate successful completion of long-sequence tasks involving real-time detection, search, and navigation toward open-vocabulary dynamic targets. To the best of our knowledge, this work presents the first operational system achieving such capabilities in unstructured environments.**

I. Introduction

In recent years, large language models (LLMs) [1] and vision models [2]–[5] have achieved revolutionary breakthroughs in the field of artificial intelligence. LLMs have significantly improved their capabilities in understanding and planning long-horizon tasks, enabling them to deeply comprehend complex contexts and generate efficient execution strategies, which brings new possibilities to task planning. Meanwhile, advances in open-vocabulary visual detection have empowered vision models to recognize and understand a diverse range of objects beyond pre-defined categories, greatly enhancing the adaptability of machine vision systems in scene understanding and object recognition. These leaps in perception and cognition lay a solid foundation for addressing complex long-horizon tasks in robotics.

Legged robot technology has evolved over decades and now demonstrates outstanding mobility in complex terrains. Their unique structural design and motion control allow them to adapt to various rugged environments, exhibiting terrain adaptability far beyond that of traditional wheeled robots. However, most current research focuses on optimizing single tasks, such as walking, jumping, climbing, and short-range navigation, lacking comprehensive consideration of complex long-horizon missions. We believe that the potential of legged robots to perform long-horizon tasks in open environments has not been fully explored, and that integrating advanced language and vision understanding with legged robot mobility is a key breakthrough for real-world applications.

In this paper, we propose LOVON (Legged Open-Vocabulary Object Navigator), an innovative integrated platform that combines the task planning capabilities of large language models, the perception abilities of open-vocabulary visual detection, and the high mobility of legged robots. The LOVON system addresses real-world challenges such as visual jitter caused by robot motion, perception blind spots, and temporary target occlusion by developing a series of dedicated solutions. In particular, we design a Laplacian Variance Filtering technique to effectively mitigate visual instability during robot movement, ensuring the accuracy and continuity of object detection. We conduct extensive experiments on multiple legged platforms (Go2, B2 and H1-2), successfully accomplishing long-horizon tasks involving real-time detection, search, and navigation toward open-vocabulary dynamic targets. To the best of our knowledge, LOVON is the first operational system to achieve such complex capabilities in unstructured environments. The main contributions of this work are summarized as follows:

- We propose the LOVON system architecture, which, for the first time, organically integrates large language models, open-vocabulary visual detection, and legged robot capabilities, providing a unified framework for the planning and execution of complex long-horizon tasks.
- We develop innovative solutions to address key real-world challenges, particularly the Laplacian Variance Filtering technique, which effectively mitigates issues such as visual jitter, blind zones, and temporary target loss, thereby enhancing the robustness of the system in dynamic environments.
- We conduct comprehensive validation on multiple legged robot platforms, and extensive experiments demonstrate the effectiveness of our system in performing open-vocabulary object search and navigation tasks in unstructured environments, paving the way for practical applications of legged robots.

II. Related Works

A. Large Language Models for Robotic Task Planning

Large language models (LLMs), such as GPT [1], PaLM [6], [7], and Llama [8], have demonstrated remarkable capabilities in natural language understanding, reasoning, and task decomposition. In the context of robotics, LLMs have been increasingly adopted for high-level task planning, instruction following, and semantic reasoning. Recent works leverage LLMs to interpret human instructions, generate action sequences, and adapt plans to dynamic environments, enabling robots to perform complex, long-horizon tasks with greater autonomy and flexibility. For example, SayCan [9] integrates LLMs with robotic affordance models to map language instructions to executable actions, while Code as Policies [10] utilizes LLMs to generate code for robot controllers directly from natural language descriptions. Despite these advances, challenges remain in grounding language to real-world robot actions [11]–[14], handling ambiguous or under-specified instructions, and ensuring robust performance in unstructured environments. Our work builds upon these foundations by integrating LLM-based planning with open-vocabulary perception and legged robot mobility, aiming to address the limitations of previous approaches in long-horizon, open-world scenarios.

B. Open-Vocabulary Visual Perception

Open-vocabulary visual perception has evolved significantly from early fixed-class object detectors like Faster R-CNN [15] and YOLO [2], [3], which were confined to recognizing pre-defined categories and struggled in open-world scenarios. Modern advancements have shifted toward models that leverage cross-modal representations, such as CLIP [16], which learns to align visual and language embeddings through contrastive pre-training on massive image-text pairs, enabling zero-shot object recognition by mapping visual features to arbitrary language descriptions. Subsequent works like Grounding DINO [17] have further enhanced this capability by integrating grounded pre-training to improve open-set detection accuracy, while DINO [4], [5] introduces denoising anchor boxes to refine end-to-end object localization for novel categories. For robotic applications, real-time performance and robustness to dynamic camera motions, such as jittering caused by legged robot locomotion or temporary target occlusions, remain critical challenges [18], [19]. Existing methods often fail to maintain detection stability in such scenarios, lacking the feedback mechanisms to adapt to the robot's motion state or environmental changes. LOVON addresses these gaps by developing specialized preprocessing techniques, like Laplacian variance filtering, to mitigate motion blur and ensure consistent visual input, while tightly integrating open-vocabulary detection with task planning and motion control for end-to-end execution in unstructured environments.

C. Legged Robot Navigation and Long-Horizon Autonomy

Legged robot navigation has advanced from low-level mobility to complex tasks, but existing approaches often focus on single-task optimization and lack integration of high-level planning for long-horizon missions [20]–[24]. While legged robots show superior terrain adaptability [25], their potential for executing sequential goals in unstructured environments remains underexplored, as traditional systems often separate perception from motion planning and struggle with dynamic target tracking .

LOVON addresses this by unifying hierarchical task decomposition with real-time motion control. An LLM-based planner breaks down long-horizon tasks, while the Language-to-Motion Model maps instructions and visual feedback to dynamic motion vectors. This allows adaptive behaviors like switching to search states when targets are lost, validated across Unitree Go2, B2 and H1-2 platforms in diverse terrains. By integrating open-vocabulary perception with legged mobility, LOVON enables autonomous robots to navigate complex environments and adapt to dynamic missions.

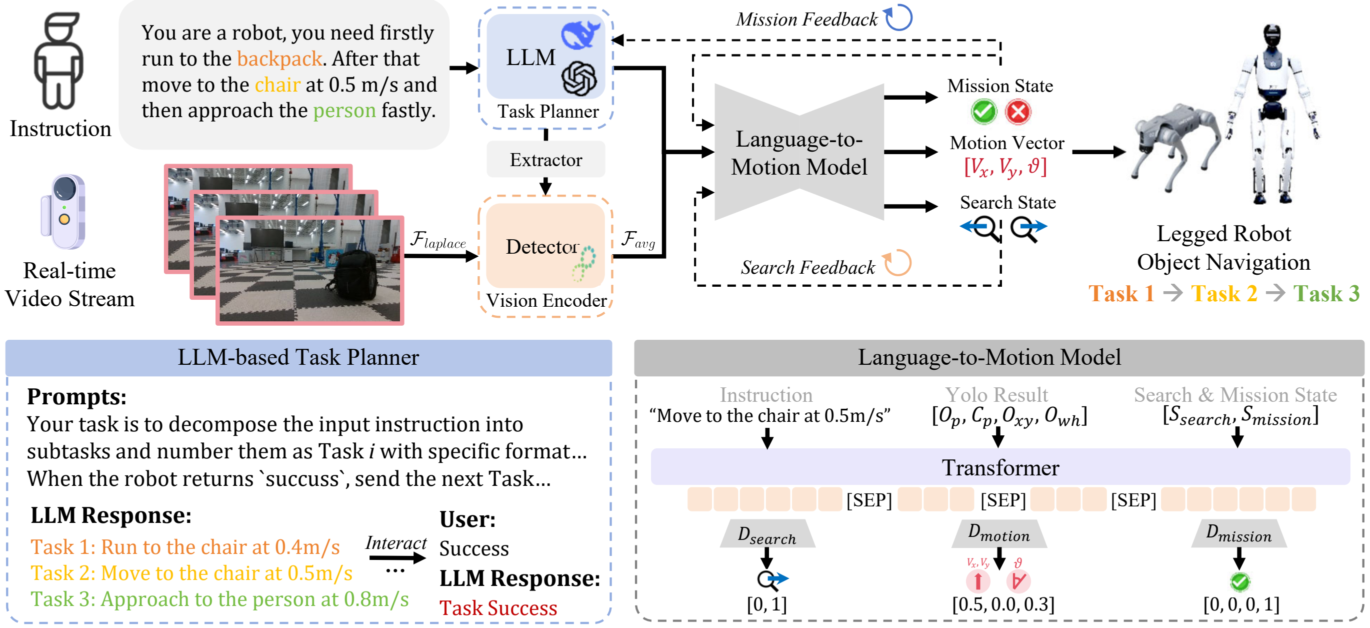

Fig. 2: **Overview of LOVON's pipeline.** First, the LLM task planner reconfigures the human's task into basic instructions, while the detection model processes the video stream using a Laplace filter. Then, the mission instructions, target object, bounding box, and states are input to the Language-to-Motion Model, which generates the robot's control vector and feedback, progressively completing all tasks.

III. Problem Formulation

Task. The task involves operating in an arbitrary open-world environment, where the robot is required to perform long-duration tasks to search for different targets. The long-duration task, T_l, is defined as a set of sub-tasks $T_l \in \{T_i | T_1, T_2, ...\}$, where each sub-task corresponds to searching for a specific target O_i. The input to the task is flexible and not fixed, allowing for varying mission objectives. The core challenge is for the robot to autonomously search for and identify different sub-goals (targets), navigating toward them at different velocities based on mission instructions. These sub-goals can vary throughout the task, requiring the robot to adapt dynamically.
Goal. Our goal is to develop a dual-system model: (I) A high-level policy that can decompose complex task instructions T_l into individual sub-tasks with concrete instruction $I_{\text{ins}} = \{I_i | I_1, I_2, ...\}$ and perform task planning; (II) A low-level policy that, based on specific sub-task instructions I_i and video stream input I_{RGB}, can generate motion vectors $V_m \in R^3$ to achieve precise motion control. The model should be adaptable across various legged robots, ensuring versatility in real-world applications.

IV. Methodology

A. Overview

The LOVON model pipeline is illustrated in Fig. 2. Initially, the LLM reconfigures the human's long-horizon task into basic mission instructions based on the mission state. These instructions are then passed to an instruction object extractor (IOE) to identify the target object. The detection model processes the captured video stream, with the input image pre-processed using a Laplace filter. Finally, the mission instruction, target object, bounding box, mission state, and search state are combined as inputs to the proposed language-to-motion model (L2MM), which generates the robot's control vector and feedback states for both the LLM and L2MM.

B. Multimodal Input Processing

LOVON integrates two pre-trained models: object detection model (e.g. [5], [17], [26], [27]) for visual input processing and LLM (e.g. [1], [28], [29]) for long-horizon task management. The object detection model takes an RGB image I_{RGB} as input and outputs detection information as follows:

$$O_m, C_p, O_{xy}, O_{wh} = f_{\text{det}}(I_{\text{RGB}}). \tag{1}$$

We use the normalized format for the detection results, with the predicted object denoted as O_m, the confidence score as C_p, and the center position of the bounding box as $O_{xy} = [x_n, y_n]$. The width and height of the bounding box are represented as $O_{wh} = [w_n, h_n]$. While the object detection model detects all visible objects, we focus on the one most relevant to the current mission. The input to the LLM model consists of the system description I_{sys}, the user's long-sequence task description T_l, and feedback from the language-to-motion model O_f. The LLM processes T_l, breaking it down into specific mission instructions $\{I_i\}$ based on historical context. This enables LOVON to execute long-sequence tasks by generating the necessary instructions to achieve mission objectives:

$$I_{\text{ins}} = f_{\text{LLM}}(I_{sys}, T_l, O_f). \tag{2}$$

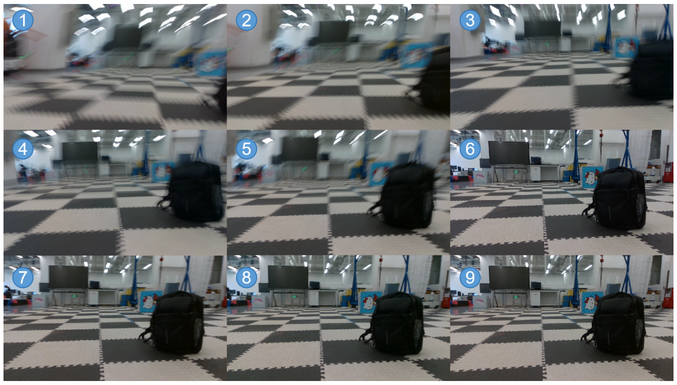

Fig. 3: **Image blurring phenomenon.** This figure shows the occurrence of image blurring in the robot view, which impacts the clarity and accuracy of the processed images.

C. Instruction Object Extractor (IOE)

The mission object extraction is treated as a sequence-to-class classification task, where the input is the mission instruction I_m from LLM, and the output is the corresponding class O_m of the object detection model. The task is modeled using a two-layer transformer followed by a perception layer as the classification head. Let the encoded instruction be $h = \text{Transformer}(I_m)$. The perception layer then outputs the predicted object class:

$$O'_m = f_{\text{IOE}}(h) \in \mathbf{C}, \tag{3}$$

where $\mathbf{C}$ represents the set of detectable classes. The IOE model is trained using cross-entropy loss.

D. Motion Blur Filtering and Bounding Box Smoothing

When the legged robot is in motion, the resulting fluctuations can cause motion blur in the captured frames, as shown in Fig. 3. The first few frames are particularly blurred due to the robot's dynamic locomotion, making them challenging for the vision model. To address this, we propose a Laplacian variance-based method for detecting and filtering motion-blurred frames. This preprocessing step improves the robustness of inputs to the object-detection-based vision-language pipeline by mitigating the effects of motion blur and distortion caused by the robot's movement and vibrations.

In particular, we first convert the RGB frame I_{RGB} to grayscale I_{gray}. We then apply the Laplacian operator to enhance high-frequency components, yielding the Laplacian response. The variance of the Laplacian response is computed to assess the clarity of the frame. If the variance is below a threshold T_{blur}, the frame is classified as blurred and replaced by the last clear frame. The threshold T_{blur} is empirically calibrated for legged robot scenarios. Additionally, we apply a moving average filter to smooth the bounding boxes from the object detection model's output, further improving stability. The performance of this frame filtering method is discussed in Sec. VII-B.

E. Language-to-Motion Model

The Language-to-Motion Model is designed using an encoder-decoder architecture. The encoder takes a sequence of inputs and generates a latent state that captures the relevant features for subsequent tasks. The input sequence consists of the following components: the previous mission instruction I_{m0}, the current mission instruction I_{m1}, the predicted object O_p, the predicted confidence C_p, the center position O_{xy}, the width and height of the normalized bounding box O_{wh}, the current mission state $S_{m_{in}}$, and the current search state $S_{s_{in}}$. These inputs are concatenated, separated by special tokens [SEP], as $I_{encoder} = \{I_{m0}, I_{m1}, O_p, C_p, O_{xy}, O_{wh}, S_{m_{in}}, S_{s_{in}}\}$ The encoder processes this sequence and outputs a latent state l_e, which is then passed to the decoder for further processing.

Next, the decoder consists of three separate heads, each designed to handle different prediction tasks:

Motion Vector Head D_{motion}. This head predicts the robot's motion vector V_m based on the latent state l_e. It is formulated as a sequence-to-vector problem, where the output is the control vector for the robot's movement. The prediction is given by:

$$V_m = D_{motion}(l_e). \tag{4}$$

Mission State Head $D_{mission}$. This head predicts the mission state S_m, which is used as feedback for the encoder to adjust future actions. The prediction is formulated as a sequence-to-number problem, where the output represents the current status of the mission. The prediction is given by:

$$S_m = D_{mission}(l_e). \tag{5}$$

Search State Head D_{search}. This head predicts the search state S_s, which indicates the robot's progress in searching for the target. It is also a sequence-to-number problem, where the output reflects the current state of the search. The prediction is given by:

$$S_s = D_{search}(l_e). \tag{6}$$

Each of these decoder heads uses a perception layer to process the latent state l_e and generate the respective outputs.

The final output of the model is a combination of the predictions from all three decoder heads:

$$O_{decoders} = \{V_m, S_{m_{out}}, S_{s_{out}}\}, \tag{7}$$

where $V_m = [v_x, v_y, \theta]$, v_x and v_y represent the robot's velocities along the x and y axes, and θ represents the angular velocity; $S_{m_{out}}$ and $S_{s_{out}}$ is the output search state and mission state, respectively. The mission state includes `success` (task completed successfully) and `running` (moving towards the detected object). The search state includes `searching_0` (searching by rotating left) and `searching_1` (searching by rotating right). The relationship between states and their corresponding motion vectors is summarized in Tab. I.

This architecture allows the model to predict motion vectors, mission states, and search states simultaneously, facilitating multi-task learning for robot control.

TABLE I: **Relationship between states and motion vector.** In the running state, v_x represents the moving speed of x axis. The rotation θ_{corr} is regulated during execution to correct the heading direction.

State S_m/S_s	Motion vector V_m
success	$[0,\ 0,\ 0\ ]$
running	$[v_x,\ 0,\ \theta_{corr}\ ]$
searching_0	$[0,\ 0,\ -0.3]$
searching_1	$[0,\ 0,\ 0.3]$

F. Loss Functions

The model is trained using different loss functions depending on the task:

Motion Vector Loss. For the motion vector head D_{motion}, we use the Mean Squared Error (MSE) loss to measure the difference between the predicted and actual motion vectors:

$$L_{MSE} = \frac{1}{N}\sum_{i=1}^{N}(V^i_{m_{pred}} - V^i_{m_{true}})^2. \tag{8}$$

Mission and Search State Loss. For the mission and search state heads $D_{mission}$ and D_{search}, we use Cross-Entropy loss to compare the predicted states with the ground truth labels:

$$L_{CE} = -\sum_{i=1}^{N} y_i \log(p_i), \tag{9}$$

where y_i is the true label, and p_i is the predicted probability for each class.

V. Robot Functional Logic for Task Execution

Once the model generates predictions, the robot follows its functional logic to execute tasks and adapt to environmental changes. The key functions that guide its behavior during task execution include:

- **Execute New Mission**: The robot compares the previous mission instruction with the current one. If they differ, the robot begins the new task.
- **Run to the Object**: Upon detecting the mission object, the robot navigates towards it based on the motion vector and the detection results.
- **Search for Lost Object**: If the robot loses track of the mission object, it automatically switches to a searching state and adjusts its motion to relocate the object.
- **Maintain the State**: The robot maintains its current state based on real-time visual inputs until a transition is triggered, ensuring consistent task execution.
- **Accomplish the Mission**: The robot monitors the mission object and, once within a success threshold (based on O_{wh}), it stops and transitions to the `success` state with $V_m = [0, 0, 0]$.

These functional rules ensure autonomous navigation, task adaptation, and robust task completion.

VI. Dataset Preparation

As illustrated in Fig. 4, the dataset generation pipeline consists of three main components:

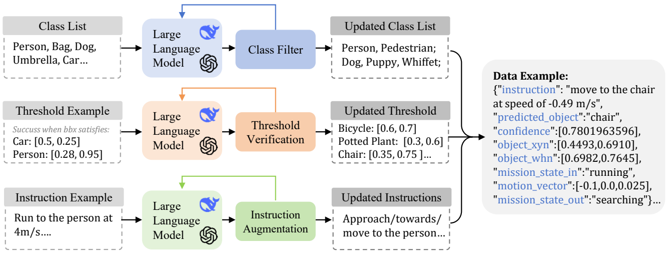
Class List
Person, Bag, Dog, Umbrella, Car...
Large Language Model
Class Filter
Updated Class List
Person, Pedestrian; Dog, Puppy, Whiffet;
Threshold Example
Success when bbx satisfies:
Car: [0.5, 0.25]
Person: [0.28, 0.95]
Large Language Model
Threshold Verification
Updated Threshold
Bicycle: [0.6, 0.7]
Potted Plant: [0.3, 0.6]
Chair: [0.35, 0.75]...
Instruction Example
Run to the person at 4m/s....
Large Language Model
Instruction Augmentation
Updated Instructions
Approach/towards/ move to the person...
Data Example:
{"instruction": "move to the chair at speed of -0.49 m/s",
"predicted_object":"chair",
"confidence":[0.7801963596],
"object_xyn":[0.4493,0.6910],
"object_whn":[0.6982,0.7645],
"mission_state_in":"running",
"motion_vector":[-0.1,0.0,0.025],
"mission_state_out":"searching"}...

Fig. 4: **Dataset Generation Pipeline.** The pipeline includes three modules: expanding object class synonyms, generating instruction variations, and adapting detection thresholds for different object categories.

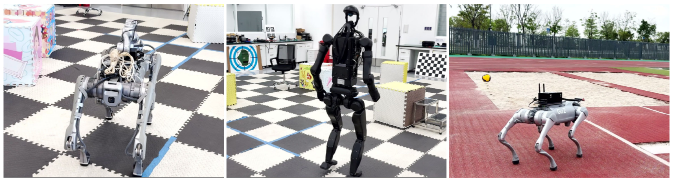

Fig. 5: **Multi-embodiment**. LOVON can be seamlessly adapted to any legged robot for precise object navigation.

Detection Class Synonym Expansion. We use an LLM to generate synonyms for the predefined object classes, enriching the object categories and improving the model's ability to generalize across different object descriptions.

Instruction Variation. To enhance the language module, we use the LLM to generate paraphrases of mission instructions. This allows the model to process diverse sentence structures while preserving core information, improving its adaptability.

Threshold Generation for Object Categories. We define success thresholds for object detection based on initial examples, then use the LLM to adapt these thresholds for other object categories, ensuring the model handles different object sizes.

During the generation process, the generated data is fed back into the LLM to refine the dataset iteratively, avoiding redundancy and improving the dataset's diversity over time.

The dataset generation process is fast and easy to expand. It takes less than 15 minutes to generate 1 million data with CPU Intel i9-12900KF.

VII. Experiment

A. Experiment Setup

Model Details. For object detection, we employ the recently developed and efficient YOLO11 [26], which offers both high performance and a lightweight architecture. As the task planner and data generation assistant, we utilize Deepseek R1 [28]. L2MM is a transformer-based model that incorporates sinusoidal positional encoding [30]. It features a dimension size of 256, with 4 transformer layers and 8 attention heads, a feedforward dimension of 1024, and a linear head layer. IOE shares the same general architecture, with key differences: a reduced feature dimension size of 64, 2 transformer layers with 4 attention heads, and a feedforward dimension of 256.

Training Settings. Our collected dataset consists of 1 million samples, which are divided into training and testing sets in a

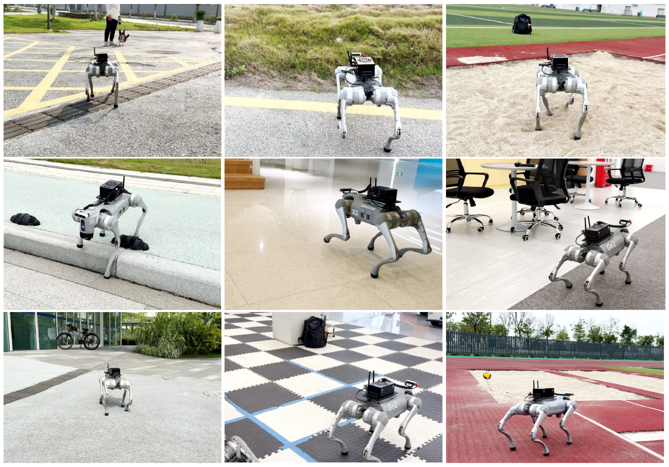

Fig. 6: **Environment adaptation.** LOVON excels in open-world object navigation, effectively adapting to a wide range of objects and environments.

4:1 ratio. We use the NVIDIA RTX 3080 Ti GPU for training. The L2MM model is trained with a dropout rate of 0.1, a learning rate of 1e-4, a batch size of 512, a maximum sequence length of 64, and a motion loss coefficient β set to 10. It is trained on a dataset containing 400,000 generated samples and undergoes 25 epochs of training using the AdamW optimizer. The total training time is approximately 1 hour. Similarly, the IOE model is trained with the same dropout rate, learning rate, batch size, and maximum sequence length. It is trained on the same 400,000 generated samples and runs for 10 epochs with the AdamW optimizer. The total training time is approximately 30 minutes.

Robot Settings. LOVON is versatile and can be applied to various legged robots. In our experiments, we evaluate three representative models: the Unitree Go2, Unitree B2, and Unitree H1, as shown in Fig. 5. For the computing platform, we utilize the Jetson Orin, while the visual platform consists of the robots' built-in cameras and the Realsense D435i camera. The experiments are conducted using the Go2 robot, with the Jetson Orin and Realsense D435i (resolution: 1280×720, frame rate: 15 FPS).

B. Performance of Motion-Blurred Frame Filtering

To investigate the impact of motion blur on object detection performance, we conduct experiments to analyze the relationship between Laplacian variance and detection confidence. We command the robot to approach a backpack, chair, or person at fixed speeds of 0.3, 0.5, or 0.7 m/s. The camera updates at approximately 15 Hz, ensuring the target remains in view. After capturing the frames, we compute the Laplacian variance for each frame and input it into the object detection model to obtain predicted confidence scores. As shown in Figure 7, there is a clear relationship between Laplacian variance and detection confidence. During the running phase, the detection model always fails to recognize the target object in many frames, despite the object remaining within view.

To address this, we adopt our proposed motion-blurred frame filtering method. By setting a blur threshold, we can filter out frames with excessive motion blur. The impact of varying blur thresholds is investigated in Figure 8, where we observe that higher thresholds result in a higher qualified frame ratio, but setting the threshold too high can lead to unnecessary rejection of valid frames. After testing, we set the threshold to $T_{blur} = 150$, which improves the qualified frame ratio by approximately 15% for all sets.

We then integrate this filtering method into our object detection pipeline. Blurred frames are excluded and replaced with the last qualified frame, and the detection confidence is smoothed using a moving average filter (MAF). As shown in the subplot of Figure 8, this integration leads to an overall 25% increase in the qualified frame rate, demonstrating the effectiveness of our motion-blurred frame filtering method.

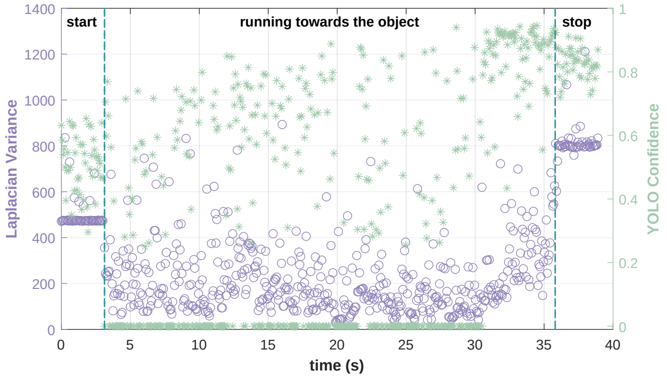

Fig. 7: **Visualization of the relationship between Laplacian variance and the confidence of the detection object.** Laplacian variance and YOLO confidence at speed 0.3 m/s towards the chair.

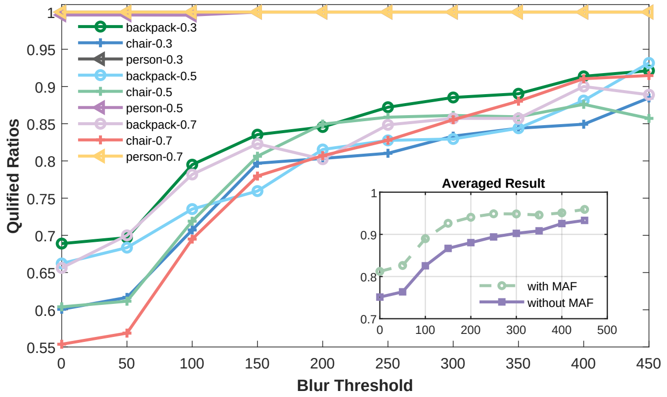

Fig. 8: **Thresholds and qualified ratios of different objects at different speeds.** The sub-figure shows the comparison of original and filtered thresholds and qualified ratios of different objects at different speeds. The filled line is the case without MAF; the dotted line is the case with MAF. The result is the average of all the 9 cases.

C. Evaluation on Simulation Benchmark

Benchmark and Evaluation Metric. In our evaluation, we follow the setup from previous works [31] under the Gym-Unreal benchmark [32], where the maximum episode length

TABLE II: **Quantitative results compared with baselines in Gym-Unreal environments.** The numbers of each cell represent the Average Episode Length (EL) and Success Rate (SR). The best results are in bold, where our LOVON achieves superior results compared with previous baselines. *DiMP uses a pre-trained video tracker which does not need additional training time.

Methods	Training time	ParkingLot	UrbanCity	UrbanRoad	SnowVillage	Mean
DiMP [34]	0 hours*	327 / 0.48	401 / 0.66	308 / 0.33	301 / 0.43	334.25 / 0.48
SARL [35]	24 hours	301 / 0.22	471 / 0.86	378 / 0.48	318 / 0.31	367.00 / 0.47
AD-VAT [36]	12 hours	302 / 0.20	484 / 0.88	429 / 0.60	364 / 0.44	394.75 / 0.53
AD-VAT+ [37]	24 hours	439 / 0.60	497 / 0.94	471 / 0.80	365 / 0.44	443.00 / 0.70
TS [38]	26 hours	472 / 0.89	496 / 0.94	480 / 0.84	424 / 0.63	468.00 / 0.83
RSPT [31]	12 hours	480 / 0.80	**500 / 1.00**	**500 / 1.00**	410 / 0.80	472.50 / 0.90
EVT [33]	1 hours	484 / 0.92	**500 / 1.00**	496 / 0.96	471 / 0.87	487.75 / 0.94
TrackVLA [23]	360 hours	**500 / 1.00**	**500 / 1.00**	**500 / 1.00**	**500 / 1.00**	**500.00 / 1.00**
LOVON (Ours)	**1.5 hours**	**500 / 1.00**	**500 / 1.00**	499 / 0.99	**500 / 1.00**	499.75 / **1.00**

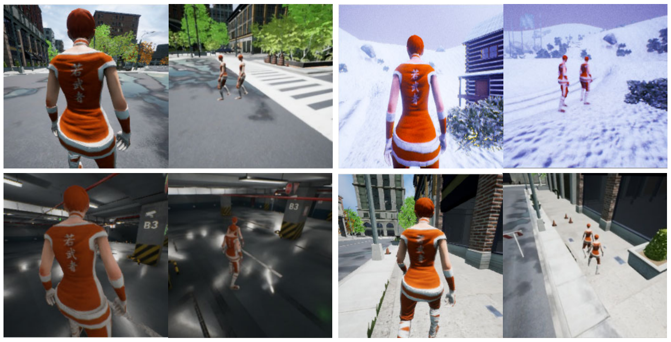

Fig. 9: **Simulation Evaluation.** We conducted extensive experiments in Gym-Unreal with four scenes: UrbanCity, SnowVillage, ParkingLot, and Urban Road.

is 500 steps. The visible region of the tracker is defined as a 90-degree fan-shaped sector with a radius of 750 cm. A tracker succeeds if it keeps the target in sight for the entire episode; failure occurs if the target exits the region for more than 50 consecutive steps. Each episode concludes with a success or failure, after which the environment and agent are reset. We evaluate performance using two metrics: 1) Episode Length (EL), which measures the average duration of episodes over 100 trials, reflecting the tracker's long-term performance; 2) Success Rate (SR), which calculates the percentage of successful episodes across 100 trials, indicating the model's robustness.

Performance Comparisons. As shown in Table II, our method, LOVON, outperforms several baseline approaches, achieving a perfect SR of 1.00 across most environments, including ParkingLot, UrbanCity and SnowVillage. Compared to EVT [33], LOVON demonstrates superior tracking performance, e.g., 500 / 1.00 vs. 484 / 0.92 in ParkingLot. Even when compared to the state-of-the-art TrackVLA [23], which achieves 1.00 SR but requires 360 hours of training, LOVON stands out with an efficient training time of just 1.5 hours, offering both high accuracy and significant efficiency.

D. Evaluation on Real-world Experiments

In real-world evaluations, LOVON demonstrates exceptional performance in four key areas. (1) First, it excels in *open-world adaptation* (Fig. 6), allowing the robot to handle a wide range of objects commonly encountered in daily life, including large objects like cars, medium-sized ones like people, and small items such as bags. This enables LOVON to seamlessly interact with various objects, regardless of their size or type, in unfamiliar environments. (2) Second, LOVON achieves *multi-goal tracking* through our LLM planner, enabling long-horizon object navigation. (Fig. 10). This capability allows the robot to efficiently track multiple objects over extended periods, even when the environment becomes more complex. (3) Third, LOVON excels in *dynamic tracking*, successfully following moving objects in dynamic environments, much like walking a dog. We tested this feature on flat roads, spiral stairs, and wild grass, and the robot reliably completed the task in these challenging conditions. (4) Finally, LOVON is *robust to disturbances*. If the targeted object is displaced or if the robot itself is disturbed (such as being kicked), the robot quickly re-localizes and continues its search. In one test, when we moved the chair and kicked the robot, LOVON still managed to approach the chair. Additionally, in a playground with sandy terrain, even when a sports ball was kicked away, the robot completed its task without difficulty.

E. Ablation Study

Ablation on the Model Parameters. We investigate the impact of model size, dataset size N_ds, motion loss weight β, and the inclusion of special tokens. We evaluate the model's performance by comparing the standard deviation σ_v and the average speed bias ϵ_v compared to the speed given in the instruction (0.40m/s). As shown in Tab. III, the baseline model shows good performance, while smaller models exhibit higher σ_v and ϵ_v, indicating they cannot effectively capture the required information. Larger models generally perform better, but the improvement in speed tracking is marginal despite the model size being much larger. Dataset size affects model stability. Models trained on smaller datasets show lower ϵ_v but higher σ_v, reflecting instability. Larger datasets

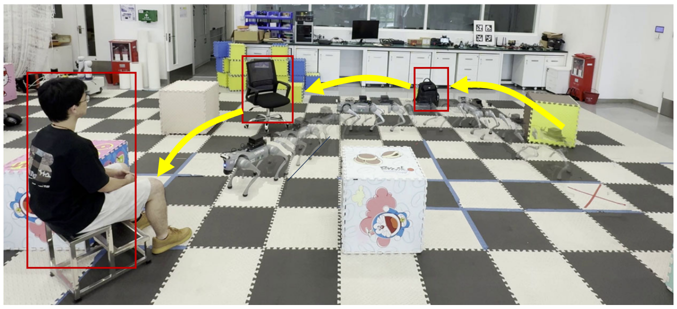

Fig. 10: **Long-Horizon Tasks with Multiple Sub-goals.** LOVON efficiently handles long-horizon object navigation by coordinating multiple sub-goals, ensuring sustained performance over extended tasks.

TABLE III: **Ablation study on model parameters.** We train all the models for 50 epochs and select the model with the lowest validation loss for evaluation.

Model Size	N_{ds} (m)	β	[SEP]	σ_v (e-4) ↓	ϵ_v (e-2) ↓
LOVON-Small	0.4	10	✓	3.74	5.74
LOVON-Large	0.4	10	✓	0.43	0.23
LOVON-Base	0.2	10	✓	3.73	0.43
	0.8	10	✓	0.40	1.49
	0.4	1	✓	4.59	7.88
	0.4	20	✓	Fail	Fail
	0.4	10	×	1.02	29.5
LOVON-Base (Best)	**0.4**	**10**	✓	**0.38**	**1.80**

improve performance but do not significantly impact real-world applicability. The motion loss weight β plays a crucial role. A smaller β=1 results in poor performance, as the model gives insufficient attention to motion loss. Conversely, a larger β=20 leads to undervaluation of state loss, causing inaccurate state inference. Finally, we find that the inclusion of the special token [SEP] is essential. The model trained without this token struggles to follow the given speed, as [SEP] helps differentiate between various input components, especially language.

Ablation on the Filter Method and the Number of States. We evaluate the impact of the number of searching states and the frame-filtering technique on the efficiency of object navigation when the target is lost. Three system configurations are tested: Case 1, with three state and no frame filtering; Case 2, with four searching states but no frame filtering; and Case 3, with four states and frame filtering. Experiments are conducted on the Go2 robot with three objects (backpack, chair, and person) at distances of 4m and 6m.

As shown in Table IV, LOVON excels in seeking the person, which is easily detected, while the backpack, which is harder to track, requires more effort. In Case 1, the robot performs inefficiently, often losing the object and requiring significant time due to motion blur. Adding a mission state in Case 2 improves efficiency by enabling the robot to search in both directions, though object loss and shaking still occur, leading to a higher number of searching circles. In Case 3, with four states and the frame-filtering technique, LOVON achieves optimal efficiency. It reduces N_s to 1 for both the backpack and chair, matching the performance with the person. Furthermore, the time to reach the target is reduced by 5 and 2 times for the backpack and chair, respectively, compared to Case 1, approaching the performance observed for the person.

TABLE IV: **Ablation study on proposed methods.** N_s: Number of Searching; T_s: Search Time (s)

Object	**States**	**Filter**	4m		6m	
			N_s ↓	T_s ↓	N_s ↓	T_s ↓
Backpack	3	×	4.05	100	5.95	178
	4	×	1.6	45.915	2.25	83.0875
	4	✓	1	22.247	1	32.5725
Chair	3	×	2.05	43.5	4.25	110
	4	×	1.05	26.2905	1.35	45.342
	4	✓	1	21.7255	1	30.551
Person	3	×	1	19.8	1	28
	4	×	1	17.6615	1	26.246
	4	✓	1	17.4735	1	26.5095

VIII. Conclusion

In conclusion, we propose LOVON, the first system that is capable of handling long-horizon tasks, adapts to unstructured environments, and achieves SOTA performance on tracking tasks. By introducing the Laplacian-Variance based frame filter and smoothing average confidence filter, the performance of the model in real-world application is significantly enhanced. Extensive real-world experiments were conducted to validate the performance of LOVON, including indoor and outdoor environments. In the future, we will continue to improve the structure of LOVON, enabling it to better integrate the latest visual language models with embodied intelligent navigation tasks.

References

[1] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[2] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A review of yolo algorithm developments," *Procedia computer science*, vol. 199, pp. 1066–1073, 2022.

[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[4] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 9650–9660.

[5] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L. M. Ni, and H.-Y. Shum, "Dino: Detr with improved denoising anchor boxes for end-to-end object detection," *arXiv preprint arXiv:2203.03605*, 2022.

[6] D. Driess, F. Xia, M. S. Sajjadi, C. Lynch, A. Chowdhery, A. Wahid, J. Tompson, Q. Vuong, T. Yu, W. Huang *et al.*, "Palm-e: An embodied multimodal language model," 2023.

[7] R. Anil, A. M. Dai, O. Firat, M. Johnson, D. Lepikhin, A. Passos, S. Shakeri, E. Taropa, P. Bailey, Z. Chen *et al.*, "Palm 2 technical report," *arXiv preprint arXiv:2305.10403*, 2023.

[8] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan *et al.*, "The llama 3 herd of models," *arXiv e-prints*, pp. arXiv–2407, 2024.

[9] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman *et al.*, "Do as i can, not as i say: Grounding language in robotic affordances," *arXiv preprint arXiv:2204.01691*, 2022.

[10] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng, "Code as policies: Language model programs for embodied control," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 9493–9500.

[11] Z. Fu, Q. Zhao, Q. Wu, G. Wetzstein, and C. Finn, "Humanplus: Humanoid shadowing and imitation from humans," *arXiv preprint arXiv:2406.10454*, 2024.

[12] J. Li, X. Cheng, T. Huang, S. Yang, R.-Z. Qiu, and X. Wang, "Amo: Adaptive motion optimization for hyper-dexterous humanoid whole-body control," *arXiv preprint arXiv:2505.03738*, 2025.

[13] Q. Zhang, P. Cui, D. Yan, J. Sun, Y. Duan, G. Han, W. Zhao, W. Zhang, Y. Guo, A. Zhang *et al.*, "Whole-body humanoid robot locomotion with human reference," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 11 225–11 231.

[14] X. Cheng, K. Shi, A. Agarwal, and D. Pathak, "Extreme parkour with legged robots," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 11 443–11 450.

[15] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.

[16] 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.

[17] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su *et al.*, "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," in *European Conference on Computer Vision*. Springer, 2024, pp. 38–55.

[18] Q. Zhang, J. Cao, J. Sun, Y. Shao, G. Han, W. Zhao, Y. Guo, and R. Xu, "Es-parkour: Advanced robot parkour with bio-inspired event camera and spiking neural network," *arXiv preprint arXiv:2503.09985*, 2025.

[19] P. Foehn, A. Romero, and D. Scaramuzza, "Time-optimal planning for quadrotor waypoint flight," *Science robotics*, vol. 6, no. 56, p. eabh1221, 2021.

[20] A.-C. Cheng, Y. Ji, Z. Yang, Z. Gongye, X. Zou, J. Kautz, E. Bıyık, H. Yin, S. Liu, and X. Wang, "Navila: Legged robot vision-language-action model for navigation," *arXiv preprint arXiv:2412.04453*, 2024.

[21] Y. Duan, Q. Zhang, and R. Xu, "Prompting multi-modal tokens to enhance end-to-end autonomous driving imitation learning with llms," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 6798–6805.

[22] J. Gao, Z. Wang, Z. Xiao, J. Wang, T. Wang, J. Cao, X. Hu, S. Liu, J. Dai, and J. Pang, "Coohoi: Learning cooperative human-object interaction with manipulated object dynamics," *Advances in Neural Information Processing Systems*, vol. 37, pp. 79 741–79 763, 2024.

[23] S. Wang, J. Zhang, M. Li, J. Liu, A. Li, K. Wu, F. Zhong, J. Yu, Z. Zhang, and H. Wang, "Trackvla: Embodied visual tracking in the wild," *arXiv preprint arXiv:2505.23189*, 2025.

[24] J. Sun, Q. Zhang, G. Han, W. Zhao, Z. Yong, Y. He, J. Wang, J. Cao, Y. Guo, and R. Xu, "Trinity: A modular humanoid robot ai system," *arXiv preprint arXiv:2503.08338*, 2025.

[25] Q. Zhang, G. Han, J. Sun, W. Zhao, C. Sun, J. Cao, J. Wang, Y. Guo, and R. Xu, "Distillation-ppo: A novel two-stage reinforcement learning framework for humanoid robot perceptive locomotion," *arXiv preprint arXiv:2503.08299*, 2025.

[26] G. Jocher and J. Qiu, "Ultralytics yolo11," 2024. [Online]. Available: https://github.com/ultralytics/ultralytics

[27] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," Springer, pp. 213–229, 2020.

[28] D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi *et al.*, "Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning," *arXiv preprint arXiv:2501.12948*, 2025.

[29] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[31] F. Zhong, X. Bi, Y. Zhang, W. Zhang, and Y. Wang, "Rspt: reconstruct surroundings and predict trajectory for generalizable active object tracking," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 3, 2023, pp. 3705–3714.

[32] e. Fangwei Zhong, Weichao Qiu, "Gym-unrealcv: Realistic virtual worlds for visual reinforcement learning," Web Page, 2017. [Online]. Available: https://github.com/unrealcv/gym-unrealcv

[33] F. Zhong, K. Wu, H. Ci, C. Wang, and H. Chen, "Empowering embodied visual tracking with visual foundation models and offline rl," in *European Conference on Computer Vision*. Springer, 2024, pp. 139–155.

[34] G. Bhat, M. Danelljan, L. V. Gool, and R. Timofte, "Learning discriminative model prediction for tracking," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 6182–6191.

[35] W. Luo, P. Sun, F. Zhong, W. Liu, T. Zhang, and Y. Wang, "End-to-end active object tracking and its real-world deployment via reinforcement learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 6, pp. 1317–1332, 2019.

[36] F. Zhong, P. Sun, W. Luo, T. Yan, and Y. Wang, "Ad-vat: An asymmetric dueling mechanism for learning visual active tracking," in *International Conference on Learning Representations*, 2019.

[37] ——, "Ad-vat+: An asymmetric dueling mechanism for learning and understanding visual active tracking," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 5, pp. 1467–1482, 2019.

[38] ——, "Towards distraction-robust active visual tracking," in *International Conference on Machine Learning*. PMLR, 2021, pp. 12 782–12 792.