2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Enhancing Vision-Based Policies with Omni-View and Cross-Modality Knowledge Distillation for Mobile Robots

Kai Li[1,2], Shiyu Zhao[2]

***Abstract*— Vision-based policies are widely applied in robotics for tasks such as manipulation and locomotion. On lightweight mobile robots, however, they face a trilemma of limited scene transferability, restricted onboard computation resources, and sensor hardware cost. To address these issues, we propose a knowledge distillation approach that transfers knowledge from an information-rich, appearance-invariant omni-view depth policy to a lightweight monocular policy. The key idea is to train the student not only to mimic the expert's actions but also to align with the latent embeddings of the omni-view depth teacher. Experiments demonstrate that omni-view and depth inputs improve the scene transfer and navigation performance, and that the proposed distillation method enhances the performance of a single-view monocular policy, compared with policies solely imitating actions. Real-world experiments further validate the effectiveness and practicality of our approach. Code will be released publicly[1].**

I. Introduction

Vision-based policies for mobile robots have a wide range of applications, from underwater exploration [1], to quadruped pursuit-evasion [2] and drone flight [3], [4]. By directly mapping raw image data to control actions, these policies eliminate the need for explicit intermediate representations such as maps and trajectories [3], and offer a more streamlined approach to vision-based robotic tasks. Among the various methods for training visuomotor policies, imitation learning (IL) has become a popular paradigm. Its high sample efficiency and straightforward formulation make it well-suited for training visuomotor policies from expert demonstrations [1], [2], [5]–[7].

However, for mobile robots, visuomotor policies face two major challenges. **First**, visuomotor policies with monocular cameras often struggle with poor generalization to unseen environments. Due to limited data diversity in both simulation and real-world settings, the trained policy may fail when exposed to new visual observations that are unseen in the training set, leading to unpredictable actions and behaviors. Even within the same environment, variations in illumination and the texture of surrounding objects can negatively affect the performance of the policy. **Second**, mobile robots are typically equipped with limited onboard computing resources, due to constraints in power consumption and mechanical structure. Visuomotor policies for mobile robots often benefit from large field-of-view (FOV) image inputs, since these deliver a large volume of visual information about the surroundings. Omnidirectional robots particularly benefit from 360° perception for improved obstacle awareness and navigation safety [8], [9]. However, constrained onboard computational resources pose challenges for real-time processing of the increased data volume of omnidirectional sensing. In addition, high-capacity image encoders are often preferred in visuomotor policies for their ability to provide high-quality visual representations, but they also require significant onboard computational resources.

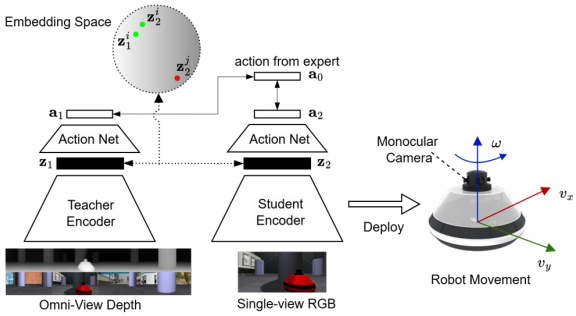

Fig. 1: The teacher policy (left) leverages appearance-invariant omnidirectional depth images generated by concatenating multi-view inputs, while the student (right) relies on a single-view RGB image. Both the teacher and the student imitate the action $\mathbf{a}$ from the expert data, with the student additionally distilling the feature embedding $\mathbf{z}$ from the teacher. Compared to the teacher, the student policy is computationally lightweight and more suitable for deployment on lightweight low-cost mobile robots.

Using depth sensors, such as RGB-D cameras [6], [7] or LiDAR [8], [10], can provide appearance-invariant depth observations, which improves the scene transfer ability of the policy. While effective, the high price and bulky size of depth sensors limits their application in low-cost, lightweight mobile robots. Alternatively, high-capacity monocular depth networks [11] can produce depth images without additional hardware and enhance the policy's scene transfer ability, but their computational demands exacerbate the existing challenge of limited onboard computation resources.

The preceding discussion about the trilemma of 1) a policy's scene transfer and safety performance, 2) onboard resource limitation, and 3) robot hardware cost, naturally leads to the following question: **is it feasible to learn a policy from a low-cost single-view RGB camera that achieves performance close to that of policies using depth images and omnidirectional views?** Motivated by this

This research work was supported by National Natural Science Foundation of China (Grant No. 62473320). (Corresponding author:Shiyu Zhao.)

[1]College of Computer Science and Technology at Zhejiang University, Hangzhou, China.
[2]School of Engineering at Westlake University, Hangzhou, China. {likai,zhaoshiyu}@westlake.edu.cn

[3] Code is at this link: https://github.com/xiaowei1015/robot-kd

question, we propose a knowledge distillation framework to enhance vision-based policies trained from imitation learning by distilling the omni-view and cross-modality information. Contributions of this paper are summarized as follows:

1) We introduce an omni-view and cross-modality knowledge distillation framework for mobile robots utilizing vision-based IL. Compared to other methods that take monocular images, policies trained using our approach achieve approximately 15% improvement in navigational success rate, an approximate 19% increase in collision-free travel distance, and a reduction in action errors.

2) In online deployment, our method eliminates the need for depth sensors or multi-camera omnidirectional systems, with onboard inference taking around 20 ms. This not only reduces the computational load on the robot but also lowers hardware costs and complexity.

3) Extensive experiments are conducted in both simulated and real-world environments. The deployment on a real-world lightweight mobile robot system with limited computation resources demonstrates the practical effectiveness and feasibility of our approach.

II. Related Works

Visuomotor Policy Learning. Visuomotor policies directly map raw pixel observations to robot actions without the need for intermediate representations such as state estimation or trajectories [3], [4], which have emerged as a promising approach in various robotic tasks such as quadruped locomotion [6], [7] and drone flight [3], [4], [12]. Due to its high sample efficiency and straightforward formulation, imitation learning has become a popular paradigm for training visuomotor policies. To enhance the performance and scene transferability of these policies, researchers have explored several methods. Some works leverage depth images [6], [7] or visual attention areas [5], [13], [14] to improve performance. The work in [15] leverages adaptive contrastive learning (ACL) to learn more discriminative visual features from monocular images. In the context of reinforcement learning (RL), the work in [12] uses cross-modality information to improve monocular vision policy performance.

Knowledge Distillation. Knowledge distillation [16] was originally proposed to transfer knowledge from a large model to a compact one. In robot policy learning, this corresponds to distilling a teacher policy into a student. While imitation learning typically trains the student to replicate the teacher's actions, recent works also distill intermediate embeddings to improve student performance. In quadruped locomotion, several studies [2], [17], [18] distill knowledge from a fully observable teacher to a student with partial observations. The student's encoder is trained to learn the teacher's latent representations, which capture information inaccessible to the student, such as an evader's predicted trajectory [2] or terrain and mechanical parameters [17], [18]. In robot manipulation [19], [20] or monocular depth estimation [21], some works [19]–[21] apply a similar principle to transfer knowledge from a teacher with information-rich input to a student with less informative input.

In this paper, we distill the robust appearance-invariant and information-rich omnidirectional depth knowledge to a compact policy that takes only a single-view RGB image for mobile robot navigation. By combining the cross-modality and omni-view distillation, we seek to improve the scene transfer and navigational success performance of the mobile robot, under the condition of limited onboard computation resources and robot hardware cost.

III. Proposed Method

A. Problem Formulation

Our work aims to learn a visuomotor policy that receives a temporal sequence of image observations and goal states, and outputs control actions. The image sequence and goal states are encoded into a joint embedding $\mathbf{z}^t$, which the policy uses to produce control actions. The policy is formulated as:

$$\mathbf{a}^t = \pi(\mathbf{z}^t), \tag{1}$$

where $\mathbf{z}^t$ denotes the latent embedding at time step t, and $\mathbf{a}^t = [v_{\mathrm{x}}^t, v_{\mathrm{y}}^t, \omega^t]$ is the 2D velocity command. The goal state $\mathbf{g}^t$ represents a 2D goal point in the robot's local coordinate frame. A key challenge with this approach is that using a monocular camera with a limited FOV can constrain the policy's scene transferability and collision avoidance capabilities, especially when trained with limited data. To address these limitations, we introduce our knowledge distillation framework in the following sections.

B. Overview

Fig. 2 provides an overview of our method. We first use an off-the-shelf expert policy π_{E} to collect demonstration data. Based on this data, we train a teacher policy π_{T} that uses omnidirectional depth images as raw input. The latent embeddings of π_{T} and actions of π_{E} are then distilled into a student policy π_{S}, which operates with single-view RGB input. Finally, π_{S} is deployed onboard to perform the task. In the following sections, we describe the architectures and training of the vision-based policies π_{T} and π_{S}, the contrastive embedding distillation, and how we obtain the expert policy π_{E}.

C. Teacher Policy Training

We first employ an off-the-shelf expert policy π_{E} to collect expert demonstrations in the form of (state, observation, action) tuples, where the state comprises the robot's global position $[x, y]$ and heading ψ, and the observation is an omnidirectional image obtained by concatenating images from the onboard multi-camera setup. Using expert data, we train the teacher policy. Since our lightweight, low-cost mobile robot is equipped with only monocular cameras, we first convert omnidirectional RGB images to depth images using DepthAnyThingV2 (DATv2) [11]. The omnidirectional depth images are then sent to an encoder network, which produces a depth image embedding. This depth embedding is concatenated with an embedding of the goal point, and linearly projected to a fused embedding $\mathbf{z}_1$. Finally, $\mathbf{z}_1$ is processed by a multilayer perceptron (MLP) network to

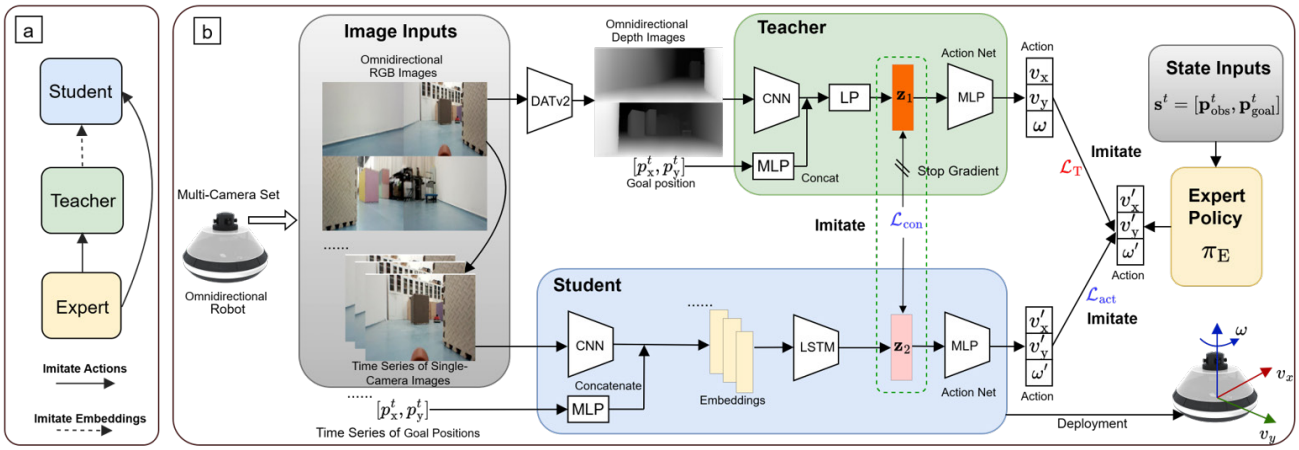

Fig. 2: Overview of the proposed method. (a) shows the knowledge transfer flow from the state-based expert, to the omnidirectional-depth-based teacher, and the RGB-based student. (b) shows the detailed pipeline of our method. In contrast with the vanilla form of IL, our method imitates both the action output and intermediate visual embeddings, which is highlighted in the dash-line boxes. LP in the teacher denotes linear projection, which is used for embedding dimension matching.

generate the final action. π_{T} is trained with a vanilla imitation loss, which minimizes the L_2 discrepancy between its output action and the expert's action,

$$\mathcal{L}_{\mathrm{T}} = \|\pi_{\mathrm{T},\theta_{\mathrm{T}}}(\mathbf{z}_1^t) - \pi_{\mathrm{E}}(\mathbf{s}^t)\|_2. \tag{2}$$

Here, the policy inputs $\mathbf{z}_1^t$ refer to the linearly projected embedding from concatenated embeddings of omnidirectional image and goal point. θ_{T} denotes the learnable parameters of π_{T}. $\mathbf{s}^t$ denotes the state inputs for π_{E}. Since there are no available image encoders designed for omnidirectional images with a large aspect ratio, we divide the omnidirectional detph image into four separate parts evenly. Each part is then processed by a shared image encoder, and the resulting embeddings are concatenated to form a final, joint embedding. The image encoder, goal state encoder and the action network are trained end-to-end with no frozen modules.

D. Student Policy Training

The student takes a time series of RGB images from a single-view monocular camera. This image sequence is processed by a shared image encoder to generate a series of image embeddings. Simultaneously, the goal point state is encoded by an MLP. At each time step, the image embedding is concatenated with the goal point embedding to form a series of fused embeddings. These fused embeddings are then passed to a long short-term memory (LSTM) recurrent module for temporal fusion. The resulting embedding $\mathbf{z}_2$, which contains both spatial and temporal information, is then fed into a final MLP to regress the action output. This approach leverages historical data to overcome the limited information of a single static image, providing the policy with a richer understanding of the robot's surroundings. The student is trained using the following loss function,

$$\mathcal{L}_{\mathrm{S}} = \lambda_0 \mathcal{L}_{\mathrm{act}} + \lambda_1 \mathcal{L}_{\mathrm{con}} \tag{3}$$

where $\mathcal{L}_{\mathrm{act}}$ is the loss for action regression and $\mathcal{L}_{\mathrm{con}}$ is the contrastive loss for aligning the feature embeddings. The hyperparameter λ_0 and λ_1 balance the two loss components. The action loss, $\mathcal{L}_{\mathrm{act}}$, minimizes the L_2 discrepancy between the action outputs of π_{S} and π_{E},

$$\mathcal{L}_{\mathrm{act}} = \|\pi_{\mathrm{S},\theta_{\mathrm{S}}}(\mathbf{z}_2^t) - \pi_{\mathrm{E}}(\mathbf{s}^t)\|_2. \tag{4}$$

The inherent limitations of a single-view RGB input, including its limited information volume and sensitivity to appearance changes, make it difficult for the student to generalize. As a result, action imitation (Eq. (4)) alone fails to ensure robust policy performance. Therefore, we introduce an additional contrastive loss $\mathcal{L}_{\mathrm{con}}$ to align the student's embeddings with those of the teacher. The contrastive loss and feature embedding alignment will be introduced in detail in Section III-E. During training of the student, the image and state encoders, LSTM, and action network are trained end-to-end without any frozen modules.

E. Knowledge Distillation via Contrastive Learning

The key part of our knowledge distillation framework is the alignment of intermediate features in the embedding space. Advances in visual representation learning and knowledge distillation [22], [23] have demonstrated that contrastive learning loss is effective in pulling together feature embeddings of similar samples while pushing apart those of dissimilar ones. This mechanism enables the learning of high-quality discriminative representations that better capture the underlying structure of the data. Based on these considerations, we use the InfoNCE contrastive loss [22] to align the teacher's embedding $\mathbf{z}_1$, with the student's embeddings $\mathbf{z}_2$. The loss function is defined as

$$\mathcal{L}_{\mathrm{con}} = \frac{1}{N}\sum_{i=1}^{N} -\log \frac{\exp\left(\mathrm{sim}(\mathbf{z}_1^i, \mathbf{z}_2^i)/\tau\right)}{\sum_{j=1}^{N} \exp\left(\mathrm{sim}(\mathbf{z}_1^i, \mathbf{z}_2^j)/\tau\right)}. \tag{5}$$

Here, τ is a temperature parameter that shapes the similarity distribution, while sim is a similarity measurement for the embeddings. N represents the batch size. In this formulation, positive pairs are embeddings corresponding to robot states that are spatially close in the global pose, denoted as $(\mathbf{z}_1^i, \mathbf{z}_2^i)$. Conversely, negative pairs are sampled from embeddings associated with spatially distant states, and represented as $(\mathbf{z}_1^i, \mathbf{z}_2^j)$. We use cosine similarity, a measurement commonly used in contrastive learning because of its scale invariance and stability, to measure embedding similarity,

$$\text{sim}(\mathbf{z}_1, \mathbf{z}_2) = \frac{\mathbf{z}_1^T \mathbf{z}_2}{\|\mathbf{z}_1\| \|\mathbf{z}_2\|}. \tag{6}$$

The InfoNCE loss pushes the embedding $\mathbf{z}_2$ of the student toward the embedding $\mathbf{z}_1$ of the teacher when sampled at spatially close states, and pushes them apart when sampled from distant poses. Since $\mathbf{z}_1$ is produced from omnidirectional view depth images, it encodes rich appearance-invariant information about the surrounding environment. By using the contrastive loss to align $\mathbf{z}_1$ and $\mathbf{z}_2$, the student's embedding is enriched with two key types of information from the teacher: 1) the surrounding environment context not visible from a single-view camera, and 2) the appearance-invariant depth information. The LSTM recurrent module in the student helps to accumulate temporal information for $\mathbf{z}_2$, which narrows the semantic gap between $\mathbf{z}_1$ and $\mathbf{z}_2$ and helps an easier and more stable embedding alignment.

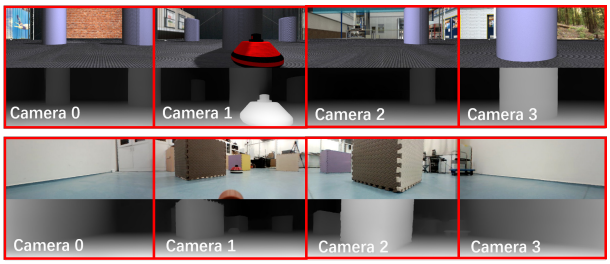

Fig. 3: Omnidirectional RGB and depth images from DATv2. The top row shows the simulation and the bottom row shows the real world. Omni-view is formed by concatenating multi-camera images. Camera 1 is used for the single-view student policy.

The omni-view and cross-modality knowledge distillation can be intuitively understood through two perspectives: **regularization** and **data augmentation**. From a regularization standpoint, the contrastive loss $\mathcal{L}_{\text{con}}$ acts as a regularizer for the imitation learning process. By adding it to the standard action loss $\mathcal{L}_{\text{act}}$, the student is not only encouraged to mimic the action of the expert but is also constrained to learn a similar, robust intermediate representation from the teacher. This prevents the student from simply overfitting to the observation-action data pairs and forces it to learn a more generalizable feature space. Alternatively, this can be viewed as a form of data augmentation. The single-view image, which is the student's input, can be considered a "cropped" or "augmented" version of the teacher's information-rich and appearance-invariant omnidirectional depth input. Our goal is to ensure that the policy achieves performance on this "augmented" input comparable to that on the original data. By using contrastive learning to align the embeddings, the policy is effectively trained on the "augmented" data, which inherently improves its representation power and performance.

F. Expert Policy for Data Generation

In this section we introduce how the expert policy π_{E} is obtained. It is worth noting that π_{E} is not limited to a particular form, for example it can be via model predictive control (MPC) [24], RL [2], [3], [15], or even human demonstrations [1]. In this work, we choose to use state-based RL to learn π_{E}, since RL enjoys higher onboard inference speed than optimization-based methods like MPC. π_{E} takes the states and outputs control commands. The state inputs are defined as $\mathbf{s}^t = [\mathbf{p}_{\text{obs}}^t, \mathbf{p}_{\text{goal}}^t]$, where $\mathbf{p}_{\text{obs}}^t$ denotes the relative positions of obstacles and $\mathbf{p}_{\text{goal}}^t$ denotes the relative goal position in the local frame. The raw state vector is first encoded by a two-layer MLP and then sent into the actor and critic network. The reward at time t is defined as $r^t = r_{\text{dist}}^t + r_{\text{obst}}^t + r_{\text{bound}}^t$. Each component of the reward is defined as,

$$\begin{aligned} r_{\text{dist}}^t &= -\alpha\,|d| \\ r_{\text{obst}}^t &= -10 \text{ if hits the obstacles} \\ r_{\text{bound}}^t &= -10 \text{ if hits the boundaries.} \end{aligned} \tag{7}$$

where α is a hyperparameter, d is the distance between the robot and goal point.

IV. Experimental Evaluations

In this section, we first introduce the experiment task, environment and the implementation details, then we evaluate scene transfer performance with embedding analysis. Finally, we evaluate the performance in the robotic task.

A. Task Description and Implementation Details

Task Description. We evaluate our knowledge distillation method on the robot navigation task. The omnidirectional robot uses images and a goal point coordinate in its local coordinate frame as input, and navigates toward the goal point while avoiding collisions. The goal point may be either static or moving according to a predefined policy. The robot uses only its onboard camera for perception and decision-making. The robot's aim is to approach the goal point and maintain desired distance and heading angle.

Data Description. The training dataset is structured as a sequence of (state, image, action) tuples. Here the state is the global position and heading of the robot. Data are collected by rolling out the state-based expert policy π_{E} in 4 different simulation environments and the real world lab environment. We build a simulation environment with ROS Gazebo [25]. The experiment area is 4 m $\times$ 4 m. The obstacles in the environment are randomly placed, and the obstacle density (obstacle area / total area) varies from 0.05 to 0.20. The images are rendered with 4 onboard RGB cameras, each with 110° horizontal FOV and 30 Hz frame rate. We calibrate the 4 cameras, which are pointed in different directions. After

calibrating each camera, we undistort the raw images based on the calibrated parameters and concatenate them to create a single omnidirectional view. The goal point information for the policy is provided by simulation or motion tracking system. In total, we collected 50k (state, image, action) tuples for policy training.

Policy Training. The state-based expert policy π_{E} is trained with the Soft Actor-Critic (SAC) algorithm [26] for 5M environment steps. The omnidirectional depth teacher distills knowledge from π_{E} and is trained with a batch size of 64. The learning rate is 1×10^{-5} for the encoder and 1×10^{-4} for the action network. Depth images are generated by DATv2 [11], each single view having a resolution of $3 \times 360 \times 180$ pixels. For the student policy, we use a batch size of 32, with the same learning rates as the teacher for the encoder (1×10^{-5}) and the action network (1×10^{-4}). The dimensions of $\mathbf{z}_1$ and $\mathbf{z}_2$ are both 1024. The weighting factor of the contrastive embedding distillation loss in Eq. (3) (λ_1) is set to 0.9 at the beginning of training and linearly decayed to 0.1. This schedule allows the policy to initially focus on embedding alignment and gradually shift its emphasis toward action learning. The weighting factor λ_0 is fixed at 1. The temperature parameter τ for contrastive learning is 0.1. The image encoder is ResNet [27], initialized from ImageNet-1K [28]. The student is trained with DAgger [29] to improve generalization power.

B. Scene Transfer Performance

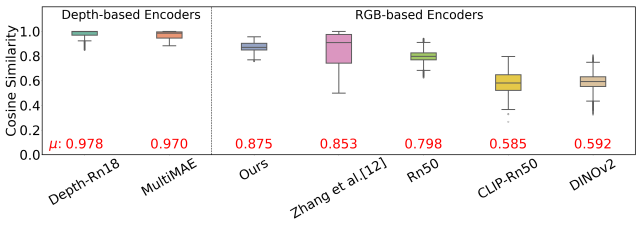

Fig. 4: Embedding similarity comparison of different image encoders across scenes. Higher similarity values indicate more consistent embeddings and stronger scene transferability. μ is the mean value of similarity.

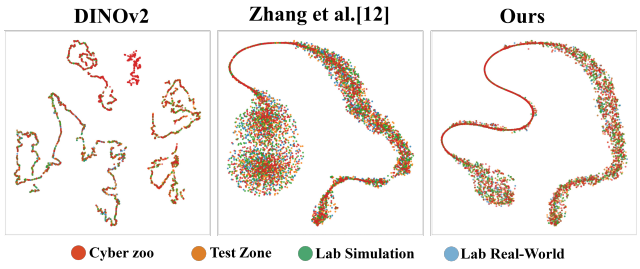

Fig. 5: t-SNE [30] visualization of visual embeddings of different scenes from DINOv2 [31], Zhang et al. [12] and ours. Different colors represent different scenes.

Visual Embedding Similarity. We evaluate the scene transfer ability of visuomotor policies by measuring the similarity of their visual embeddings across different scenes.

TABLE I: White rows show action error results for methods without embedding distillation, using different input modalities and camera views. Gray rows show results with the proposed knowledge distillation method. *Finetuned* indicates that the encoder is further trained on our custom robot dataset, while non-finetuned encoders use frozen weights pretrained on a large-scale public dataset.

Method	Modality	View	Finetuned	AE Train ↓	AE Test ↓
Resnet18 [27]	RGB	Omni	✗	0.0038	0.0226
Resnet18 [27]	RGB	Single	✗	0.0051	0.0734
Resnet50 [27]	RGB	Omni	✗	0.0027	0.0093
Resnet50 [27]	RGB	Single	✗	0.0042	0.0112
DINOv2 [31]	RGB	Omni	✗	0.0030	0.0097
DINOv2 [31]	RGB	Single	✗	0.0049	0.0129
CLIP Rnet [33]	RGB	Omni	✗	0.0031	0.0138
CLIP Rnet [33]	RGB	Single	✗	0.0052	0.0143
MAE [34]	RGB	Single	✗	0.0036	0.0090
Resnet18 [27]	Depth	Omni	✓	**0.0012**	**0.0037**
Resnet18 [27]	Depth	Single	✓	0.0020	0.0050
MultiMAE [35]	RGB-D	Single	✗	0.0029	0.0091
Ours-Rnet18	RGB	Single	✓	0.0030	0.0070
Ours-Rnet50	RGB	Single	✓	0.0021	0.0067

To do this, we use distinct simulation environments that share the same layout, meaning the positions and sizes of obstacles, as well as the robot's start and goal positions, are identical. The only variable is the environment's visual appearance. We use the average cosine similarity of the embeddings to quantify this transfer ability. Ideally, with identical scene layouts, test policies and initial robot states, the visual embeddings at the same position should be highly similar across different scenes. As shown in Fig. 4, both the depth-image and RGB-D-based encoders (MultiMAE [35]) achieve high embedding similarities. Among RGB-based encoders, our knowledge distillation framework attains the highest mean embedding similarity, indicating more consistent features across different scenes. We also use t-SNE [30] to visualize embeddings from different scenes. Fig. 5 shows the visualization results. The embeddings of ours are more consistent across different scenes compared with other RGB-based encoders. Moreover, the embedding distribution aligns with the robot's spatial movements and preserves meaningful spatial variation in the input images, ensuring that distinct observations are represented by distinct embeddings. In contrast, embeddings from pretrained models (e.g. DINOv2) scatter without clear structure.

Visual Attention Maps. We use the Full Gradient method [32] to generate visual attention maps. In Fig. 6, we compare the attention maps from encoders trained with and without our knowledge distillation method. The results demonstrate that our policy, trained with knowledge distillation, learns to focus its attention on task-relevant features, aligning more closely with human intuition. In contrast, the policy without knowledge distillation is easily distracted by irrelevant features in the RGB input, leading to a less robust representation.

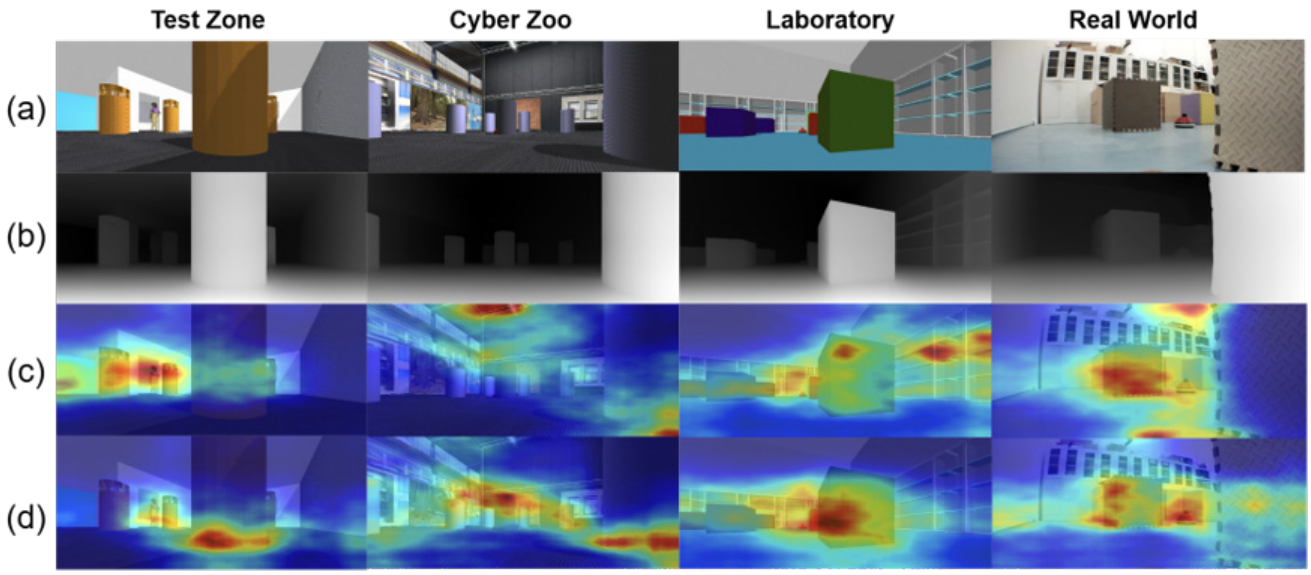

Fig. 6: The rows show (a) RGB images, (b) depth images, (c) Full Gradient [32] (FG) attention maps from a policy trained directly on RGB, and (d) FG attention maps from a policy trained with our knowledge distillation method. The attention maps in row (c) are distracted by irrelevant features. In contrast, the maps in row (d) align more with human intuition, consistently focusing on obstacles when they are nearby. The columns show different simulation and real-world scenes. The encoder backbones of (c) and (d) are both Resnet50.

C. Mobile Robot Task Performance

Action Error. We perform comparison and ablation studies using Action Error (AE) to quantify the discrepancy between expert and student outputs. Table I reports the AE of different visual encoders under various training conditions without embedding distillation (white rows), and with embedding distillation (gray rows). For visual encoders that are not fine-tuned, we use pre-trained weights from large-scale public datasets. For a fair comparison, all policies incorporate temporal information using an LSTM with the same sequence length of 5 steps. The results confirm that a larger volume of input information (i.e. omni-view input) and the depth modality are advantageous for policy performance. Specifically, the data show that using omnidirectional depth images produces the lowest AE. **When the input modalities are the same**, an omni-view input generally lowers AE in both training and testing scenes compared to single-view setups, which indicates improved policy performance. Similarly, **when the types of input view are the same**, using depth as the input modality leads to a lower AE, suggesting enhanced imitation performance. The policy trained with our omni-view, cross-modality distillation achieves a lower AE than other RGB-based encoders, approaching the performance of omni-view depth input. This demonstrates the effectiveness of our framework in leveraging omni-view and cross-modality information to enhance visuomotor policies.

Navigational Performance. Table II shows the results of the robot navigation experiment. AE is the action error in test scenes. The success rate (SR) is defined as the proportion of experiment episodes in which the robot reaches the goal within 0.3 m without collision. The moving distance (MD) is the distance the robot moves in one episode before collision with obstacles or boundaries. We perform 80 independent runs in simulation (20 for each scene) and 20 runs in the real-world for each experiment. The onboard computer is an NVIDIA Jetson Orin NX, featuring 6 ARM cores, 16 GB memory, 25 W power consumption, and up to 100 TFLOPS of computing performance. The onboard model is accelerated with TensorRT with FP16 precision. From the results in Table II, we can see that using our knowledge distillation shows the best AE, SR and MD performance among RGB-based encoders, with a 23% increase in SR and 20% in MD. Since the robot is equipped with RGB cameras only, we use DATv2 to generate depth images onboard. Although using depth as input shows stronger performance in simulation, in the real-world, the additional latency (25.7 ms for single-view and 55.0 ms for omni-view) for depth image inference is undesirable for real-time control. Moreover, the CPU and GPU load introduced by a separate high-capacity depth estimation module further increases the onboard burden (with a 39% GPU load increase and 19% CPU increase). High-capacity models such as DINOv2 takes longer inference time (more than 130 ms), which is not suitable for onboard running. In contrast, with our distillation method, the policy performance gets enhanced without relying on depth estimation module, which is particularly beneficial for mobile robots with limited onboard computation resources. Fig. 7 shows typical robot trajectories under different input modalities and views. Policies using raw RGB images collide with obstacles due to poor scene transfer. In contrast, depth input or RGB with our distillation method enables safe navigation and target tracking. Single-view depth input (blue

TABLE II: Results for the robot navigation task. All onboard models are accelerated with NVIDIA TensorRT using FP16 precision. For a fair comparison, all policies incorporate temporal information using an LSTM with the same sequence length of 5. The table reports Action Error (AE), Success Rate (SR), and Moving Distance (MD). The onboard time is presented as the mean, with the 99% percentile shown in parentheses. The latency of the raw image is around 30 ms. For methods using depth as input, the inference time of DATv2 must be included in the total latency.

Method	Modality	View	Finetuned	AE ↓	SR (%) ↑	MD (m) ↓	#Parameters	Encoder Time (ms) ↓	DATv2 Time (ms)
Resnet18 [27]	RGB	Single	✗	0.0739	31.0	4.08 ± 0.34	11.7M	16.51 (26.64)	-
Resnet18 [27]	RGB	Omni	✗	0.0479	40.0	5.19 ± 0.42	11.7M	21.36 (29.54)	-
Resnet18 [27]	Depth	Single	✓	0.0050	69.0	8.28 ± 0.33	11.7M	10.15 (17.62)	25.69 (34.88)
Resnet18 [27]	Depth	Omni	✓	0.0037	74.0	8.94 ± 0.31	11.7M	12.20 (19.90)	54.97 (82.20)
Resnet50 [27]	RGB	Single	✗	0.0229	38.0	5.35 ± 0.21	23.5M	24.10 (33.81)	-
Resnet50 [27]	RGB	Omni	✗	0.0115	49.0	7.09 ± 0.30	23.5M	45.34 (58.78)	-
Resnet50 [27]	Depth	Single	✓	0.0069	70.0	8.87 ± 0.14	23.5M	15.52 (26.89)	25.69 (34.88)
Resnet50 [27]	Depth	Omni	✓	0.0033	80.0	8.90 ± 0.28	23.5M	42.78 (60.30)	54.97 (82.20)
CLIP Rnet [33]	RGB	Single	✗	0.0165	40.0	5.13 ± 0.67	38.3M	24.98 (32.76)	-
MultiMAE [35]	RGB-D	Single	✗	0.0130	49.0	5.92 ± 0.55	87.1M	43.97 (51.26)	25.97 (31.17)
DINOv2 [31]	RGB	Single	✗	0.0158	34.0	4.85 ± 0.61	86.0M	132.45 (190.01)	-
MAE [34]	RGB	Single	✗	0.0095	50.0	6.25 ± 0.32	86.6M	35.68 (50.20)	-
Ours-Rnet18	RGB	Single	✓	0.0070	67.0	8.08 ± 0.21	11.7M	16.85 (26.61)	-
Ours-Rnet50	RGB	Single	✓	0.0067	72.0	8.49 ± 0.25	23.5M	24.10 (31.45)	-

trajectory) can still cause minor collisions when obstacles leave the field of view.

TABLE III: Comparison with other methods in the literature that focus on enhancing the performance of visuomotor policies.

Method	AE (Train)↓	AE (Test)↓	SR %↑	MD (m)↑
RoboSaGA [5]	0.0088	0.0301	47.0	5.59 ± 0.51
VISARL [14]	0.0095	0.0295	40.0	5.02 ± 0.47
Zhang et al. [12]	0.0054	0.0082	62.0	8.30 ± 0.22
Ours-Rnet50	**0.0021**	**0.0067**	**72.0**	**8.49 ± 0.25**

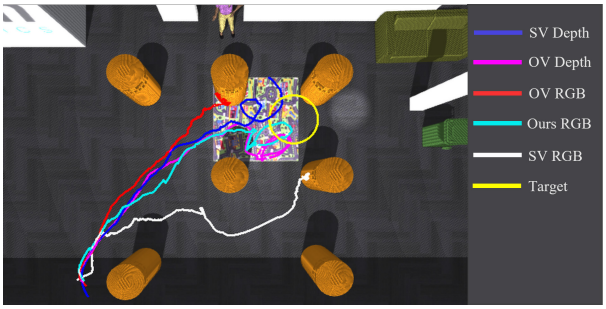

Fig. 7: Typical experiment trajectories of robots under policies trained with different strategies and input modalities. The target is moving in a circle (Yellow trajectory). SV denotes single-view image input, and OV denotes omni-view image input.

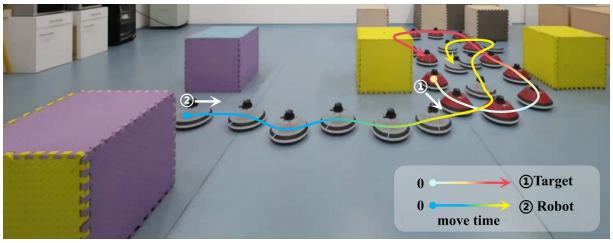

Fig. 8: Robot trajectories in real-world experiment. The white is the robot and the red is the dynamic target.

D. Comparisons With Other Methods

We compare our knowledge distillation method with open source methods aimed at improving scene transfer and overall performance in visuomotor policies. RoboSaGA [5] and VISARL [14] leverage visual attention maps to enhance visuomotor policies, while Zhang et al. [12] employs contrastive learning to improve the image encoder's representation power. Table III shows the results. While both our method and [12]'s use cross-modality feature learning, ours show better performance in our task. The reason for this is that our method jointly distills both intermediate embeddings and action outputs, enabling the visual encoder to be optimized using gradients jointly derived from the task-relevant action loss and the feature-relevant contrastive loss. In contrast, [12] trains the cross-modality encoder solely via contrastive learning and freezes the encoder during downstream RL policy learning, which limits the visual encoder's ability to adapt to the specific task. This phenomenon is consistent with the recent findings of Wang et al. [36], which concluded that treating the image encoder as part of the policy and performing end-to-end training results in better performance. Frozen pretrained image encoders cannot adapt their learned features to the specific visual characteristics required for the downstream action learning task, which can lead to suboptimal performance. In addition, [12] does not consider omni-view inputs, which limits policy performance on robots with omnidirectional mobility. Other methods [5], [14] conduct enhancement only in monocular vision domain, without the appearance-invariant depth and omni-view information. Ours outperforms them in terms of AE, SR and MD, with a SR increase of 16.1% compared with the strongest baseline. This proves the contribution of our method in improving the scene transfer and navigational performance for mobile robots.

V. Conclusions

In this paper, we propose an omni-view cross-modality knowledge distillation framework to enhance vision-based policies for mobile robots. By distilling intermediate features and action outputs from a teacher policy trained with omni-view depth observations, a monocular policy can inherit rich spatial representations, leading to improved transferability and navigation safety. Extensive simulation and real-world experiments demonstrate that our method outperforms pre-trained encoders and alternative enhancement approaches, achieving approximately 20% higher success rates.

References

[1] T. Manderson, J. C. G. Higuera, S. Wapnick, J.-F. Tremblay, F. Shkurti, D. Meger, and G. Dudek, "Vision-based goal-conditioned policies for underwater navigation in the presence of obstacles," in *Proceedings of Robotics: Science and Systems (RSS)*, 2020.

[2] A. Bajcsy, A. Loquercio, A. Kumar, and J. Malik, "Learning vision-based pursuit-evasion robot policies," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9197–9204, IEEE, 2024.

[3] T. Wu, Y. Chen, T. Chen, G. Zhao, and F. Gao, "Whole-body control through narrow gaps from pixels to action," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2025.

[4] I. Geles, L. Bauersfeld, A. Romero, J. Xing, and D. Scaramuzza, "Demonstrating agile flight from pixels without state estimation," in *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

[5] Z. Zhuang, R. Wang, N. Ingelhag, V. Kyrki, and D. Kragic, "Enhancing visual domain robustness in behaviour cloning via saliency-guided augmentation," in *Conference on Robot Learning (CoRL)*, 2024.

[6] X. Cheng, K. Shi, A. Agarwal, and D. Pathak, "Extreme parkour with legged robots," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11443–11450, IEEE, 2024.

[7] Z. Zhuang, Z. Fu, J. Wang, C. Atkeson, S. Schwertfeger, C. Finn, and H. Zhao, "Robot parkour learning," in *Conference on Robot Learning (CoRL)*, PMLR, 2023.

[8] Z. Wang, T. Ma, Y. Jia, X. Yang, J. Zhou, W. Ouyang, Q. Zhang, and J. Liang, "Omni-perception: Omnidirectional collision avoidance for legged locomotion in dynamic environments," in *Proceedings of Conference on Robot Learning (CoRL)*, PMLR, 2025.

[9] H. Xu, Y. Zhang, B. Zhou, L. Wang, X. Yao, G. Meng, and S. Shen, "Omni-swarm: A decentralized omnidirectional visual–inertial–uwb state estimation system for aerial swarms," *IEEE Transactions on Robotics*, vol. 38, no. 6, pp. 3374–3394, 2022.

[10] G. Xu, T. Wu, Z. Wang, Q. Wang, and F. Gao, "Flying on point clouds with reinforcement learning," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025.

[11] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, and H. Zhao, "Depth anything v2," in *Proceedings of Conference on Neural Information Processing Systems (NIPS)*, vol. 37, pp. 21875–21911, 2024.

[12] Y. Zhang, J. Xiao, and M. Feroskhan, "Learning cross-modal visuomotor policies for autonomous drone navigation," *IEEE Robotics and Automation Letters*, vol. 10, pp. 5425 – 5432, 2025.

[13] C. Liu, Y. Chen, M. Liu, and B. E. Shi, "Using eye gaze to enhance generalization of imitation networks to unseen environments," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 5, pp. 2066–2074, 2021.

[14] A. Liang, J. Thomason, and E. Bıyık, "Visarl: Visual reinforcement learning guided by human saliency," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2907–2912, IEEE, 2024.

[15] J. Xing, L. Bauersfeld, Y. Song, C. Xing, and D. Scaramuzza, "Contrastive learning for enhancing robust scene transfer in vision-based agile flight," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5330–5337, IEEE, 2024.

[16] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[17] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "RMA: Rapid motor adaptation for legged robots," in *Proceedings of Robotics: Science and Systems (RSS)*, 2021.

[18] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science Robotics*, vol. 5, no. 47, p. eabc5986, 2020.

[19] C. Acar, K. Binici, A. Tekirdağ, and Y. Wu, "Visual-policy learning through multi-camera view to single-camera view knowledge distillation for robot manipulation tasks," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 691–698, 2023.

[20] W. Chen and N. Rojas, "Trakdis: A transformer-based knowledge distillation approach for visual reinforcement learning with application to cloth manipulation," *IEEE Robotics and Automation Letters*, vol. 9, no. 3, pp. 2455–2462, 2024.

[21] K. Han, D. Muhle, F. Wimbauer, and D. Cremers, "Boosting self-supervision for single-view scene completion via knowledge distillation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9837–9847, 2024.

[22] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of International Conference on Machine Learning (ICML)*, pp. 1597–1607, PMLR, 2020.

[23] Y. Tian, D. Krishnan, and P. Isola, "Contrastive representation distillation," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2020.

[24] C. E. Luis and A. P. Schoellig, "Trajectory generation for multiagent point-to-point transitions via distributed model predictive control," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 375–382, 2019.

[25] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 3, pp. 2149–2154, IEEE, 2004.

[26] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, *et al.*, "Soft actor-critic algorithms and applications," *arXiv preprint arXiv:1812.05905*, 2018.

[27] 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 (CVPR)*, pp. 770–778, 2016.

[28] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255, 2009.

[29] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pp. 627–635, JMLR Workshop and Conference Proceedings, 2011.

[30] L. V. d. Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[31] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, *et al.*, "DINOv2: Learning robust visual features without supervision," *Transactions on Machine Learning Research Journal*, pp. 1–31, 2024.

[32] S. Srinivas and F. Fleuret, "Full-gradient representation for neural network visualization," *Proceedings of Conference on Neural Information Processing Systems (NIPS)*, vol. 32, 2019.

[33] 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 *Proceedings of International Conference on Machine Learning (ICML)*, pp. 8748–8763, PMLR, 2021.

[34] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16000–16009, 2022.

[35] R. Bachmann, D. Mizrahi, A. Atanov, and A. Zamir, "MultiMAE: Multi-modal multi-task masked autoencoders," in *Proceedings of European Conference on Computer Vision (ECCV)*, 2022.

[36] R. Wang, Z. Zhuang, S. Jin, N. Ingelhag, D. Kragic, and F. T. Pokorny, "Feature extractor or decision maker: Rethinking the role of visual encoders in visuomotor policies," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2025.