This paper has been accepted for publication at the
IEEE International Conference on Robotics and Automation (ICRA), Yokohama 2024 ©IEEE

Contrastive Learning for Enhancing Robust Scene Transfer in Vision-based Agile Flight

Jiaxu Xing*, Leonard Bauersfeld*, Yunlong Song, Chunwei Xing, and Davide Scaramuzza

***Abstract*— Scene transfer for vision-based mobile robotics applications is a highly relevant and challenging problem. The utility of a robot greatly depends on its ability to perform a task in the real world, outside of a well-controlled lab environment. Existing scene transfer end-to-end policy learning approaches often suffer from poor sample efficiency or limited generalization capabilities, making them unsuitable for mobile robotics applications. This work proposes an adaptive multi-pair contrastive learning strategy for visual representation learning that enables zero-shot scene transfer and real-world deployment. Control policies relying on the embedding are able to operate in unseen environments without the need for finetuning in the deployment environment. We demonstrate the performance of our approach on the task of agile, vision-based quadrotor flight. Extensive simulation and real-world experiments demonstrate that our approach successfully generalizes beyond the training domain and outperforms all baselines. Video: https://youtu.be/4A4YyPgEWD8**

I. Introduction

The ideal mobile robot could perform any task, anywhere, and without relying on external sensors—just like us humans. However, even the most advanced autonomous mobile robots only surpass human performance in well-controlled conditions on a narrowly defined task [1]. Many recent works on multi-task learning [2]–[4] address one part of this problem, namely how a robot can learn to perform different tasks in one environment. However, a mobile robot's utility also strongly depends on its ability to perform the same task outside of a known environment in the real world. To possess such *scene transfer* capabilities, the robot requires a robust perception and control system that operates consistently across different environments.

In terms of the well-established paradigm [6]–[8] to compress the perception of the environment into an intermediate representation and then use this for the action (control) network, the challenge can be rephrased as follows: how can we learn a task-related perception embedding that is invariant to the environment in which the task needs to be performed? Such a perception embedding removes the need for finetuning the action network in a specific environment and thus is an important step towards the zero-shot deployment of a mobile robot in a new environment and the real world.

* These authors contributed equally. The authors are with the Robotics and Perception Group, Department of Informatics, University of Zurich, and Department of Neuroinformatics, University of Zurich and ETH Zurich, Switzerland (http://rpg.ifi.uzh.ch). This work was supported by the European Union's Horizon 2020 Research and Innovation Programme under grant agreement No. 871479 (AERIAL-CORE), and the European Research Council (ERC) under grant agreement No. 864042 (AGILEFLIGHT).

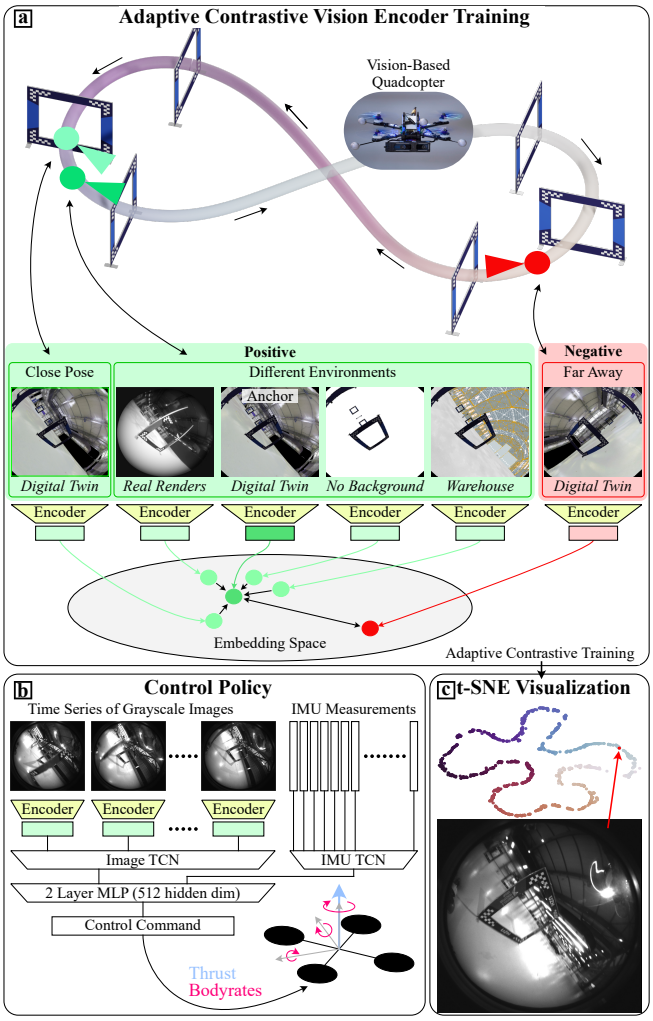

Fig. 1. We train a vision encoder using our proposed adaptive contrastive learning strategy. Positive examples are sampled from different environments and nearby points and negative examples are from far-away track segments, as shown in (a). Then the action network controlling the robot has access to a history of vision embeddings as well as IMU measurements. The action net predicts the control commands for the mobile robot, e.g. thrust and body rate commands for a quadrotor, as shown in (b). Our adaptive contrastive learning embeds the images into a self-consistent and scene-invariant feature space, shown via a t-SNE [5] visualization in (c).

For agile aerial vehicles, it is especially difficult to learn a vision representation that enables robust scene transfer. These robots can move freely in 3D space, and thus, even in the same environment, the observations can be vastly different. Additionally, there are few constraints on how the environment will look as both nearly textureless or feature-rich indoor and outdoor environments are possible. The action network must control a system that moves at high

speed and requires constant action to not crash into obstacles or the environment. Finetuning the policy in the real world is difficult due to safety concerns and because finetuning requires the policy to perform well enough to collect data and not crash the robot immediately.

These challenges present a stark contrast to prior work on scene transfer. One family of works addresses scene transferability for vision-based object manipulation and shows impressive robustness to the clutter in the scene [9], [10] and background invariance [11]. In the context of mobile robot navigation, end-to-end vision-based control has primarily been explored in settings where the training and testing environment are the same (e.g., simulation [12]–[15]) or as a way to bridge the sim2real gap. The latter can be done using a handcrafted intermediate representation that is task- and environment-specific [1], [16], [17].

However, the existing approaches mentioned do not effectively tackle the challenge of scene transfer. Achieving vision-based end-to-end control for mobile robot navigation that generalizes beyond the training environment remains an unsolved problem. The significant challenge lies in obtaining a suitable visual embedding for action learning, with proposed solutions encompassing both world models [10] and task-specific models [11], [18]. One key aspect is ensuring that the learned visual embedding is both environment-agnostic and task-relevant. Addressing this concern, contrastive learning emerges as an ideal approach for aligning the acquired embeddings. Contrastive learning serves as a robust framework for generating representations that adhere to similarity constraints within a dataset, typically structured around pairs of similar and dissimilar samples. Notably, contrastive learning has showcased promising performance on various robot learning tasks [18], [19].

In this work, we present an adaptive contrastive learning approach (shown in Fig. 1) to train a vision encoder that outputs a task-related but environment-independent embedding and that generalizes to the real world without finetuning. This advance is made possible by using i) a multi-pair contrastive loss combined with a novel formulation that replaces the constant temperature [20] approach with an adaptive temperature that captures the similarity of the contrastive pairs. ii) a novel contrastive pair sampling strategy to enhance the embedding similarity among various scenes, hence forcing the network to learn task-related features. The vision encoder is paired with a small action net, trained via privileged imitation learning [21] purely in simulation. We present extensive simulation and real-world experiments, comparing our approach with world-model-type approaches and task-specific vision encoders. Our approach outperforms all baselines and transfers to unseen environments where the action net has never been trained.

II. Related Work

A. End-to-end policy learning

End-to-end sensorimotor policy learning aims to directly predict actions from raw sensor measurements, such as RGB [22] or depth images [23], [24]. Contrary to traditional methods, such as [25], end-to-end approaches often operate without the requirement of environmental understanding [26]–[28], accurate state estimation [29], [30], or motion planning [30], [31]. These approaches rely less on handcrafted features or system redundancy. Regarding system control, recent work demonstrates the advantages of end-to-end approaches over classical optimal control in autonomous racing [32]. However, end-to-end approaches commonly face challenges related to poor sample efficiency and limited ability for scene transfer [33], and existing approaches try to overcome this problem using handcrafted intermediate representations [16], [17], [34] to enable robust scene transfer.

B. Visual pre-training for robotics

Recently, visual pre-training has shown great success for various robotics tasks using learned action-centric representations [34]–[36]. Pretrained large-scale models, such as CLIP [37] and masked autoencoder [38], have demonstrated significant potential in terms of generalization across various tasks for robotics [34], [39]. These models are trained in a supervised fashion on large datasets and have promising capabilities in adapting to diverse scenarios and tasks [39]. At the same time, self-supervised contrastive pretraining [19], [40] and contrastive reinforcement learning (RL) have also shown notable advancements in enhancing the performance of policy learning [41], [42]. In the context of this work, especially two works on contrastive RL are important, as our contrastive loss is inspired by them. In [43], [44], a contrastive RL strategy was used to ensure that the representations of future states are closer than the representations of random states. Specifically, the positive and negative pairs are chosen based on the state trajectory of the manipulator. In the following, we refer to this strategy as *State CL*. In [18] an image-based contrastive RL approach is proposed where the positive pairs are generated as augmentations of the same image. This image-based contrastive RL approach demonstrated a significant enhancement in sample efficiency for action policy learning. We refer to this as *Aug CL*.

It is noteworthy that while the aforementioned approaches have demonstrated impressive achievements in various robotics tasks, none of them have shown significant capabilities for scene transfer in the domain of mobile robot navigation. Our proposed contrastive learning approach focuses on enhancing scene transfer capabilities by using an adaptive contrastive learning approach and novel contrastive pair formation strategy. The resulting scene-consistent representation makes it particularly well-suited for mobile robot navigation tasks.

III. Methodology

Our goal is to learn a sensorimotor policy capable of directly mapping raw onboard images and proprioception (e.g. IMU measurements) into quadcopter control commands (thrust and body rates [45]). The design choice to incorporate IMU measurements for agile robot control is driven by the

high sensor frequency at around 200Hz [16]. To evaluate the performance of the proposed method, we chose the task of autonomous, vision-based drone racing [46]. The goal is to navigate through a sequence of predefined racing gates in minimum time. Fig. 1a exemplarily shows the *figure 8* racetrack along with the onboard camera images from various simulated and real environments. Drone racing is a challenging control task due to the high accelerations, non-linear aerodynamic effects, and low-latency requirements.

We follow the paradigm of using a vision-encoder which maps the perception input into an embedding space (Fig. 1a). The control policy then uses a time-series of such embeddings together with proprioceptive measurements as an input to output the control commands (Fig. 1b). We propose a ResNet-based vision encoder trained via multi-pair adaptive contrastive learning. The encoder is trained on images of a drone flying around the race track in multiple environments and the images require a pose (position and orientation) annotation. Additionally, we use the task-specific track progress to disambiguate potential self-intersections of the tracks. Collecting the images required to train the vision encoder can be done in many ways, e.g. in simulation, manually or automatically, and the pose annotation can be obtained from the simulation, motion-capture systems or structure-from-motion (SfM) approaches [47], [48]. After the vision encoder has been trained it is frozen.

To train the action network we use privileged imitation learning [49] approach in a DAgger setting [50]. First, a state-based teacher policy is trained via deep reinforcement learning [51]. We use the imitation learning approach [50], [52], to distill knowledge from the teacher policy into the student policy. The vision-based student policy exclusively relies on image embeddings from the vision encoder and raw IMU measurements.

A. Adaptive Contrastive Learning

First, we will discuss our proposed contrastive training pipeline in detail and then provide a brief overview of the state-based teacher training and vision-based student training. Given a set of n different scenes $\mathcal{S} = \{S_1, \dots, S_n\}$, the primary objective is to train a vision encoder $\phi_{\boldsymbol{\theta}}(\cdot)$ with learnable parameters $\boldsymbol{\theta}$ that demonstrate both *intra-scene* and *inter-scene* consistency.

1) Intra-scene consistency: Within a scene S_i, consider a reference (subsequently referred to as *anchor*) image I_{anc} at a certain track progress p_{anchor} (i.e. relative completion of the lap). The objective is to learn a representation that demonstrates similarity for nearby (w.r.t. track progress) images and dissimilarity for far images (see Fig. 1a, *embedding space*). This ensures a descriptive embedding for the subsequent action policy learning process. The objective can be formulated as a contrastive learning problem, where the soft-nearest neighbor (SNN) loss [20] $\mathcal{L}_{\text{snn}}$ is minimized for each datapoint within a batch:

$$\mathcal{L}_{\text{snn}} = -\log \frac{\sum_{p\in\mathcal{P}} \exp\left[-\text{sim}(\phi_{\boldsymbol{\theta}}(I_{\text{anc}}), \phi_{\boldsymbol{\theta}}(I_{\text{p}}))/\tau\right]}{\sum_{i\in\mathcal{N}\cup\mathcal{P}} \exp\left[-\text{sim}(\phi_{\boldsymbol{\theta}}(I_{\text{anc}}), \phi_{\boldsymbol{\theta}}(I_{\text{i}}))/\tau\right]} \quad (1)$$

Here $\mathcal{P}$, $\mathcal{N}$ refer to the sets of positive/negative samples which are samples of close/far progress (w.r.t. the anchor sample) within the track. I_p, I_n represent the images perceived at the corresponding progress. $\text{sim}(a, b)$ represents the cosine similarity between two embedding vectors a and b. The additional parameter τ, commonly referred to as *temperature*, can be used to guide the learned distribution in the embedding space. In [53] this effect is analyzed in detail, and the authors find that the selection of temperature significantly affects the quality of the learned representations.

We take these findings as an inspiration and augment the contrastive learning loss to a novel, adaptive temperature formulation which replaces the constant τ. Given the nature of most mobile robot navigation tasks, we leverage pose information to judge how similar positive and negative pairs are. We define a similarity measurement for the robot pose $\boldsymbol{x} = [\boldsymbol{p}_{\text{norm}}, \boldsymbol{q}]$, where $\boldsymbol{p}_{\text{norm}}$ is the [0, 1] normalized position, $\boldsymbol{q}$ is the rotation quaternion, and $\alpha(\cdot,\cdot)$ calculates the angle between two quaternions:

$$\tau(\boldsymbol{x}_1, \boldsymbol{x}_2) = \tau(\boldsymbol{p}_1, \boldsymbol{q}_1, \boldsymbol{p}_2, \boldsymbol{q}_2) = |\boldsymbol{p}_1 - \boldsymbol{p}_2|_1 \cdot \frac{\alpha(\boldsymbol{q}_1, \boldsymbol{q}_2)}{\pi} \quad (2)$$

Intuitively, if two poses are extremely similar, the adaptive τ ensures any dissimilarity in their embeddings is heavily weighted in eq. (1).

2) Inter-scene consistency: Apart from self-consistency within a scene, we want to improve the consistency across different scenes, as this property is key for scene transfer. For the drone racing example at hand, the embedding for the same pose should not depend on the background where the drone flies. To achieve this, we sample random scene changes and image augmentations during training (see Fig. 1a, different environments). The positive and negative pairs are randomly sampled from any scene in $\mathcal{S}$ and augmented through contrast and brightness changes. The vision encoder should enforce the images perceived at similar progress from different environments to have similar embeddings. More specifically, the network training is formulated as:

$$\underset{\boldsymbol{\theta}}{\text{argmin}} \underset{X_{\text{scene}}, X_{\text{aug}}}{\mathbb{E}} \left[\mathcal{L}_{\text{snn}}(\boldsymbol{\theta};\ I(x_{\text{scene}}, x_{\text{aug}}))\right] \quad (3)$$

X_{scene}, X_{aug} are two random variables controlling scenes and data augmentation randomization, and $x_{\text{scene}}, x_{\text{aug}}$ are the corresponding realizations. In our setting, we fully randomized the image in multiple scenes uniformly for all the contrastive pairs and applied an augmentation with a fixed probability.

B. Action Net Learning

1) Teacher policy learning: The state-based teacher policy takes the state observations $\mathbf{o}_t$ as input and outputs control commands. At each timestep, the policy has access to an observation from the environment which contains the current robot state (position, orientation, velocity, body rates) and the relative position to the next waypoint to be passed. The teacher policy is trained using model-free reinforcement learning approach [54], and we adapt the reward formulation based on [51], [55]. The reward r_t at time t is formulated as

$$r_t = r_t^{\text{prog}} + r_t^{\text{perc}} - r_t^{\text{act}} - r_t^{\text{br}} + r_t^{\text{pass}} - r_t^{\text{crash}} \quad (4)$$

where r_t^{prog} rewards progress towards the next gate to be passed [56], r_t^{perc} encodes perception awareness by adjusting the vehicle's attitude such that the optical axis of its camera points towards the next gate's center, r_t^{act} penalizes action changes from the last time step, r_t^{br} penalizes body rates and consequently reduces motion blur, r_t^{pass} is a binary reward that is active when the robot successfully passes the next gate, r_t^{crash} is a binary penalty that is only active when a collision happens, which also ends the episode. The reward components are formulated as follows:

$$\begin{aligned} r_t^{\text{prog}} &= \lambda_1(d_{\text{Gate}}(t-1) - d_{\text{Gate}}(t)) \\ r_t^{\text{perc}} &= \lambda_2 \exp(\lambda_3 \cdot \delta_{cam}^4) \\ r_t^{\text{act}} &= \lambda_4 \|\boldsymbol{a}_t - \boldsymbol{a}_{t-1}\| \\ r_t^{\text{br}} &= \lambda_6 \|\boldsymbol{\omega}_{\mathcal{B},t}\| \\ r_t^{\text{pass}} &= 10.0 \quad \text{if robot passes the next gate} \\ r_t^{\text{crash}} &= -5.0 \quad \text{if robot crashes (gates, ground)} \end{aligned} \tag{5}$$

Here $d_{\text{Gate}}(t)$ denotes the distance from the robot's center of mass to the center of the next gate to pass, δ_{cam} is the angle between the camera's optical axis and the direction towards the center of the next gate. $\boldsymbol{a}$ represents the control command, and $\boldsymbol{\omega}$ the body rate. $\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5$ are different hyperparameters.

2) Student policy learning: As shown in Fig. 1b the vision-based student policy takes a sequence (history length is 12 timesteps) of IMU measurements (180 Hz) $\mathbf{m}_t$ and visual embeddings (30 Hz) $\mathbf{e}_t$ as input. In contrast to approaches that jointly train vision and action module [18], the vision encoder is frozen after the contrastive pretraining. We use two separate Temporal Convolutional Networks (TCN) [57] to encode the series of vision embeddings and IMU measurements. Then the output features from two TCNs are concatenated and fed into a two-layer MLP, which outputs the actions. Note that using two TCNs allows different update rates for IMU measurements and images. We define an action loss $\mathcal{L}_A$ (action error) that is the mean square error between the outputs of the teacher policy and the student policy

$$\mathcal{L}_A(\boldsymbol{\theta}) = \|\boldsymbol{\pi}_{\text{student}}(\boldsymbol{\theta};\ \boldsymbol{e}_t, \boldsymbol{m}_t) - \boldsymbol{\pi}_{\text{teacher}}(\boldsymbol{o}_t)\| \tag{6}$$

IV. Experiments

To achieve successful scene transfer for mobile robot navigation, we require a vision encoder that enables the action net to learn a meaningful sensorimotor policy that maps the environment observations to control commands. First, we briefly give implementation details and describe the training setup. Then we evaluate the performance of the contrastive encoder. Subsequently, we evaluate the performance of mobile robot control policies in a set of simulated and real-world experiments at the drone racing task.

A. Implementation Details and Model Training

1) Image data for vision encoder training: The images for vision encoder training are collected by rolling out the state-based teacher in four simulation environments (see Fig. 1a). These environments consist of a simple warehouse scene, a digital twin of the testing environment, an empty scene, and one realistic digital twin rendered using Blender [58]. To replicate the sensor configuration on our quadrotor platform, the images are rendered using a grayscale fisheye camera model (double sphere model [59]). In total, we collected 12000 images from three rollouts in four environments.

2) Vision encoder training: We perform contrastive encoder training on the collected images as described in section III-A. The vision encoder consists of a backbone based on ResNet50 [60] and a reprojection layer [61] to generate 128-dimensional embedding from images of size 224×224. For training, there are 24 negative samples and 12 positive samples associated with each image. For each anchor image, the positive samples are uniformly sampled within $[0, 0.05] \cup [0.95, 0, 1]$ relative track progress, while negative samples are uniformly sampled within $[0.4, 0.6]$ relative track progress. The normalized track progress is a metric to describe the lap completion and is in the range of $[0, 1]$. For the model training, we employ the Adam optimizer [62] with a linearly decayed learning rate, initially set at 1e-4.

3) Student policy learning: The student policy is trained via DAgger [50]. For imitation learning, we use a batch size of 512, and the training converges typically after collecting 500K data samples after around 100 epochs (about 6 h on a desktop PC). We apply a linear decay on the learning rate, starting from 1e-3 and decreasing to 1e-5 at 50 epochs and remaining unchanged for the remainder of the training.

B. Vision Embedding

1) Intra-scene consistency: For the action net to successfully learn a sensorimotor policy, we require descriptive, task-related, and informative vision embeddings. For the task of vision-based robots racing through a given track, we hypothesize that the intra-scene consistency is a suitable proxy measure: if two images are sampled closely to each other along the track (with a similar track progress), we would expect a very high similarity in the embedding space. Similarly, the further apart (in terms of progress along the racetrack), the lower the similarity should be. Intuitively, this facilitates action learning as close-by states require similar actions, whereas far-away states require different actions. For evaluation, we use 1000 images sampled around the teacher policy rollout trajectories.

Fig. 2 shows the embedding cosine similarity as a function of track progress difference, where 0% progress difference means the same location along the trajectory and 50% progress difference is the opposite side of the track. The reported similarities are averaged values around the whole track. We can see that pretrained 'world-model' like methods (dash-dotted lines) show very high similarities independent of track progress. This makes sense as most models are trained on very large datasets containing millions of different scenes—being on the opposite side of a racetrack in the same environment, thus only causing a small change in the embedding space.

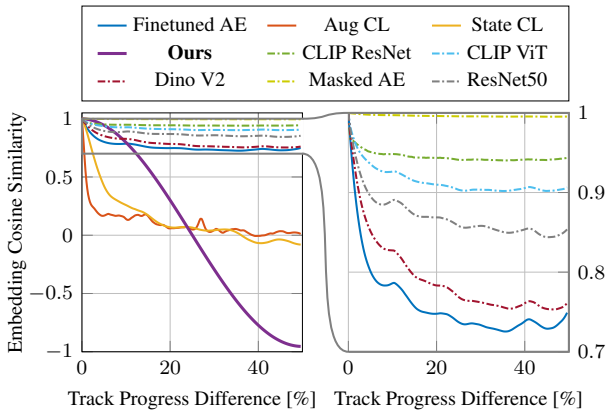

Fig. 2. This figure shows the intra-consistency of the different methods. Solid lines represent methods that have been finetuned and dash-dotted lines represent pretrained methods. One can clearly see that pretrained methods show a much more similar embedding, even at opposite ends of the track (50%). Our proposed method produces an embedding that is very distinctive, as far points on the track have very dissimilar embedding.

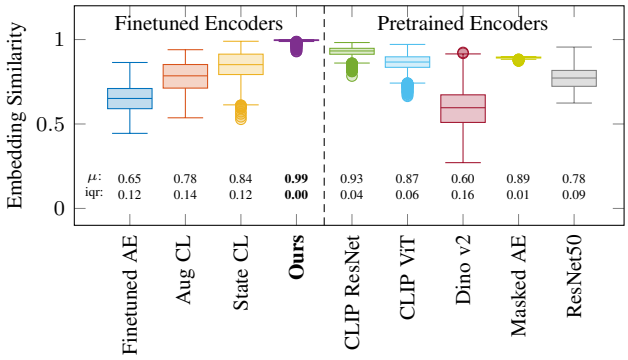

Fig. 3. This figure shows the inter-consistency of the different methods. This is a measure of how close the embeddings from different environments are together (higher is better). Our method clearly outperforms the baselines, as a high inter-consistency is desired while ensuring a low intra-consistency for far apart samples. The numbers below each bar indicate the mean embedding similarity μ and inter-quartile-range iqr.

Methods finetuned on task-specific images from the racing environment (solid lines in Fig. 2) generally show much less similarity for far away samples while retaining high similarity for close samples. Due to using 12 positive samples and augmenting the images at training time, our method achieves a much larger difference in embedding similarity than any other method.

2) Inter-scene consistency: In the previous section we looked at intra-scene consistency as a proxy measure for the descriptiveness of the vision embedding. To enable scene transfer, however, the embedding does not only have to be descriptive but also invariant to the scene in which the task is performed. In Fig. 3, we analyze how similar embeddings are if the robot's pose remains unchanged but the environment changes. Again, 1000 poses around the track are sampled, and the cosine similarity of the embeddings is compared for each pose.

First, one notices that there is no clear trend between pretrained and finetuned methods. Our method clearly outperforms all other methods with both the lowest interquartile range (iqr) and the highest average similarity of 0.99. Other methods like the masked autoencoder (Masked AE) or CLIP also perform very well with high similarity and low variance. To understand the true performance, however, we have to combine the results of intra-scene and inter-scene consistency: while the masked AE shows great results in the inter-scene consistency, it shows no distinctiveness in the intra-scene consistency, meaning that it just always generates similar embeddings. The contrastive learning baselines (Aug CL and State CL) performed well in terms of intra-scene distinctiveness but showed very high variance in the inter-scene consistency. Only our method is able to provide a task-related descriptive embedding while being scene-invariant.

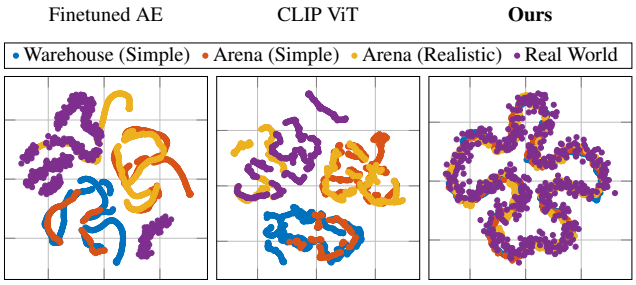

Fig. 4. t-SNE Visualizations of the embeddings for different environments. The illustration graphically summarizes what can be read quantitatively from Fig. 2 and Fig. 3: only our approach produces distinctive, task-related embeddings that are similar across all environments.

C. Embedding visualization

We apply a t-SNE dimensionality reduction [5] to visualize the embeddings from four distinct environments in a 2D space, as depicted in Figure 4. This visualization shows graphically what the previous section described: Firstly, our approach effectively encodes images within each environment into coherent and continuous representations. Secondly, for our approach, the vision embeddings across multiple environments are extremely similar (on top of each other). For both types of baselines, we observe somewhat coherent clustering (lines of the same color are mostly continuous, but the environments are vastly different from each other.

D. Action Learning

Our proposed vision encoder has achieved very good results in terms of consistency. We now turn towards the action learning to analyze which embedding is suitable for agile, vision-based flight. In this section, we use a fixed vision encoder that has been trained on all simulation environments (warehouse, no background, digital twin), but has never seen real-world data.

In the first set of simulation and real-world experiments, we record a rollout of the teacher policy and then evaluate the action error of the student policy w.r.t. the teacher policy. We use the teacher policy to generate the rollout to ensure fairness and evaluate all approaches on the same data. The student policies are trained via DAgger [50] in simulation and have only seen the warehouse and no-background environment. The digital twin, as well as real-world data, is not used during action-net training. For drone dynamics, we use a very accurate dynamics model [64], [65]. The evaluation is performed in four different settings: in the

TABLE I

Action Error of the different methods evaluated in four different settings: the training environment, a simple unseen test environment, a highly realistic (unseen) blender render using a digital twin of the real world, and real-world flight data. Our method outperforms all baselines.

Methods	**Custom Dataset**	**AE Train (Simple)**	**AE Test (Simple)**	**AE Test (Realistic)**	**AE Real World**
CLIP ResNet	✗	0.016	0.043	0.142	0.116
CLIP ViT [37]	✗	0.019	0.057	0.154	0.115
MAE [38]	✗	0.019	0.063	0.217	0.136
Resnet [60]	✗	0.018	0.071	0.112	0.087
Dino V2 [63]	✗	0.019	0.062	0.124	0.098
Autoencoder	✓	0.015	0.026	0.068	0.088
Aug CL [18]	✓	0.011	0.036	0.091	0.078
State CL [43]	✓	0.009	0.031	0.069	0.073
Ours w/o pose	✓	0.010	0.022	0.041	0.043
Ours	✓	**0.008**	**0.012**	**0.039**	**0.036**

training environment, in the unseen digital-twin environment (both with the low-fidelity renderer and the realistic blender renderer), and on data collected in the real world. The results are summarized in Table I. In all test cases, our method achieves the lowest action error, with the difference becoming more pronounced when scene transfer matters: on the unseen digital twin, we outperform the baselines by a factor of two. In the real world, all methods degrade significantly, but our method still outperforms the next-best by 25%.

In a second set of experiments, we evaluate the closed-loop control performance of all methods in the same simulation environments as already presented above. We do not show real-world results as our current network is not running fast enough on the onboard computer Nvidia Jetson TX2 to stabilize the vehicle (requires at least 100Hz control frequency). The results are summarized in Table II. We define the evaluation metrics success rate (SR) as the ratio of successful tests over the total number of tests, where each of the 64 tests starts from a different position that is uniformly sampled from the predefined box, and the number of test steps covers two complete laps. Each test is counted as a "success" only when the robot survives the whole testing episode without crashing. In the training environment, all methods perform adequately, and the differences in terms of action error (AErr) and SR are small. When the methods are deployed on the unseen digital twin of our testing facility, all methods except ours fail while we retain at least a 44% SR in the highly realistic simulator. Meanwhile, we also demonstrate that our approach could generalize to unseen random scene/brightness/blurriness changes during closed-loop deployment. We refer the reader to the accompanying video to visualize these experiments.

E. Beyond Drone Racing

To demonstrate that our approach can be applied beyond vision-based agile flight tasks, we ablate the importance of pose annotations for the contrastive encoder training. Instead of privileged pose information, we use only time-series information during the encoder training. This kind of information is available for all sequential tasks and equally applies to drone racing as it does to open a door. Additionally, instead of the adaptive temperature shown in Eq. (2), a constant one of 0.5 is used, as our adaptive temperature formulation is based on the poses of the individual samples as shown in Eq. (2). The results shown in Table I ('Ours w/o pose') demonstrate that the privileged information helps the overall performance of our approach. Nevertheless, relying only on time-series information to train the contrastive encoder, we still outperform all baselines and are (on real-world data) only 10% worse in terms of action error compared to a privileged encoder training. This strongly suggests our contrastive training pipeline has implications for general mobile robotics tasks beyond drone racing.

V. Conclusion

We presented a contrastive learning-based approach to facilitate scene transfer for vision-based agile flight. By using a novel adaptive loss formulation, the network was able to learn a task-specific embedding that is highly descriptive and consistent across many scenes. These properties set it apart from large 'world-model type', pre-trained encoders like CLIP [37] or Dino v2 [63]. We evaluated our method on seen and unseen environments and found that it outperforms all baselines in terms of action error and closed-loop experiment success rates. Lastly, we showed evidence that our contrastive training pipeline can not only be used for drone racing but generalized to vision-based sequential tasks. Future work could include more prior knowledge of the physical world to further enhance policies' robustness and safety [66].

TABLE II

Success rate (SR), Action Error, and Average Gate Passing (AGP) of the different methods when evaluated in closed-loop operation for the drone racing task. When the control policy is deployed in an environment it has been trained in, all methods perform very well. When deployed in an environment that is unseen during the training of the action net, only our approach is able to perform the task, highlighting its ability to transfer.

Methods	**Custom Dataset**	**Train (Simple)**			**Test (Simple)**			**Test (Realistic)**		
		SR [%]↑	Act Err↓	AGP↑	SR [%]↑	Act Err↓	AGP↑	SR [%] ↑	Act Err↓	AGP↑
CLIP Rn50 [37]	✗	93.8	0.014	9.42±0.20	0.0	0.078	2.85±0.48	0.0	0.194	1.55±0.16
CLIP ViT [37]	✗	95.3	0.015	9.29±0.43	0.0	0.052	2.93±0.29	0.0	0.203	1.46±0.16
MAE [38]	✗	89.1	0.017	9.39±0.28	0.0	0.043	3.02±0.27	0.0	0.210	1.32±0.14
Resnet [60]	✗	95.3	0.013	9.51±0.16	0.0	0.072	2.83±0.46	0.0	0.112	2.08±0.33
Dino V2 [63]	✗	95.3	0.015	9.05±0.19	0.0	0.058	2.98±0.30	0.0	0.153	1.54±0.15
Autoencoder	✓	**98.4**	0.015	9.31±0.38	0.0	0.136	1.67±0.19	0.0	0.139	1.67±0.19
Aug CL [18]	✓	96.8	0.011	9.51±0.44	0.0	0.091	3.11±0.31	0.0	0.124	1.91±0.21
State CL [43]	✓	**98.4**	**0.010**	**9.82**±0.32	0.0	0.040	3.68±0.32	0.0	0.082	2.45±0.12
Ours w/o pose	✓	96.8	0.013	9.57±0.42	48.4	0.035	5.01±0.23	31.3	0.073	4.43±0.10
Ours	✓	**98.4**	0.012	9.65±0.20	**64.1**	**0.023**	**7.42**±0.42	**43.8**	**0.047**	**6.03**±0.56

REFERENCES

[1] E. Kaufmann, L. Bauersfeld, A. Loquiercio, M. Müller, V. Koltun, and D. Scaramuzza, "Champion-level drone-racing with deep reinforcement learning," *Nature*, 2023.

[2] C. Fifty, E. Amid, Z. Zhao, T. Yu, R. Anil, and C. Finn, "Efficiently Identifying Task Groupings for Multi-Task Learning," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 27 503–27 516.

[3] M. Shridhar, L. Manuelli, and D. Fox, "Perceiver-actor: A multi-task transformer for robotic manipulation," in *Conference on Robot Learning*. PMLR, 2022, pp. 785–799.

[4] R. Rahmatizadeh, P. Abolghasemi, L. Bölöni, and S. Levine, "Vision-based multi-task manipulation for inexpensive robots using end-to-end learning from demonstration," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 3758–3765.

[5] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

[6] B. Zhou, P. Krähenbühl, and V. Koltun, "Does computer vision matter for action?" *Science Robotics*, vol. 4, no. 30, p. eaaw6661, 2019.

[7] A. Sax, J. O. Zhang, B. Emi, A. Zamir, S. Savarese, L. Guibas, and J. Malik, "Learning to navigate using mid-level visual priors," in *Proceedings of the Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, L. P. Kaelbling, D. Kragic, and K. Sugiura, Eds., vol. 100. PMLR, 30 Oct–01 Nov 2020, pp. 791–812.

[8] D. Driess, I. Schubert, P. Florence, Y. Li, and M. Toussaint, "Reinforcement Learning with Neural Radiance Fields," in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35. Curran Associates, Inc., 2022, pp. 16 931–16 945.

[9] A. Zeng, S. Song, K.-T. Yu, E. Donlon, F. R. Hogan, M. Bauza, D. Ma, O. Taylor, M. Liu, E. Romo, *et al.*, "Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching," *The International Journal of Robotics Research*, vol. 41, no. 7, pp. 690–705, 2022.

[10] S. Reed, K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-maron, M. Giménez, 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," *Transactions on Machine Learning Research*, 2022, featured Certification.

[11] M. Schwarz, A. Milan, A. S. Periyasamy, and S. Behnke, "Rgb-d object detection and semantic segmentation for autonomous manipulation in clutter," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 437–451, 2018.

[12] P. R. Wurman, S. Barrett, K. Kawamoto, J. MacGlashan, K. Subramanian, T. J. Walsh, R. Capobianco, A. Devlic, F. Eckert, F. Fuchs, L. Gilpin, P. Khandelwal, V. Kompella, H. Lin, P. MacAlpine, D. Oller, T. Seno, C. Sherstan, M. D. Thomure, H. Aghabozorgi, L. Barrett, R. Douglas, D. Whitehead, P. Dürr, P. Stone, M. Spranger, and H. Kitano, "Outracing champion Gran Turismo drivers with deep reinforcement learning," *Nature*, vol. 602, no. 7896, pp. 223–228, 2022.

[13] Y. Song, H. Lin, E. Kaufmann, P. Dürr, and D. Scaramuzza, "Autonomous overtaking in gran turismo sport using curriculum reinforcement learning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 9403–9409.

[14] F. Fuchs, Y. Song, E. Kaufmann, D. Scaramuzza, and P. Dürr, "Super-human performance in gran turismo sport using deep reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4257–4264, 2021.

[15] M. Müller, V. Casser, N. Smith, D. L. Michels, and B. Ghanem, "Teaching UAVs to Race: End-to-End Regression of Agile Controls in Simulation," in *Computer Vision – ECCV 2018 Workshops*, L. Leal-Taixé and S. Roth, Eds. Cham: Springer International Publishing, 2019, pp. 11–29.

[16] E. Kaufmann, A. Loquercio, R. Ranftl, M. Müller, V. Koltun, and D. Scaramuzza, "Deep drone acrobatics," in *Proceedings of Robotics: Science and Systems*, Corvalis, Oregon, USA, July 2020.

[17] X. Xiao, B. Liu, G. Warnell, and P. Stone, "Motion planning and control for mobile robot navigation using machine learning: A survey," *Auton. Robots*, vol. 46, no. 5, p. 569–597, jun 2022.

[18] M. Laskin, A. Srinivas, and P. Abbeel, "Curl: Contrastive unsupervised representations for reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2020, pp. 5639–5650.

[19] N. Messikommer, Y. Song, and D. Scaramuzza, "Contrastive initial state buffer for reinforcement learning," in *2024 International Conference on Robotics and Automation (ICRA)*, 2024.

[20] N. Frosst, N. Papernot, and G. Hinton, "Analyzing and improving representations with the soft nearest neighbor loss," in *International conference on machine learning*. PMLR, 2019, pp. 2012–2020.

[21] D. Chen, B. Zhou, V. Koltun, and P. Krähenbühl, "Learning by cheating," in *Conference on Robot Learning (CoRL)*, 2019.

[22] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1334–1373, 2016.

[23] A. Loquercio, E. Kaufmann, R. Ranftl, M. Müller, V. Koltun, and D. Scaramuzza, "Learning high-speed flight in the wild," *Science Robotics*, vol. 6, no. 59, p. eabg5810, 2021.

[24] A. Agarwal, A. Kumar, J. Malik, and D. Pathak, "Legged locomotion in challenging terrains using egocentric vision," in *Conference on Robot Learning*. PMLR, 2022, pp. 403–415.

[25] J. Xing, G. Cioffi, J. Hidalgo-Carrió, and D. Scaramuzza, "Autonomous power line inspection with drones via perception-aware mpc," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 1086–1093.

[26] B. Sun, J. Xing, H. Blum, R. Siegwart, and C. Cadena, "See yourself in others: Attending multiple tasks for own failure detection," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 8409–8416.

[27] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto, "Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1366–1373.

[28] A. Rosinol, M. Abate, Y. Chang, and L. Carlone, "Kimera: an open-source library for real-time metric-semantic localization and mapping," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2020.

[29] M. Hanlon, B. Sun, M. Pollefeys, and H. Blum, "Active visual localization for multi-agent collaboration: A data-driven approach," in *2024 International Conference on Robotics and Automation (ICRA)*, 2024.

[30] C. Forster, M. Pizzoli, and D. Scaramuzza, "Svo: Fast semi-direct monocular visual odometry," in *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2014, pp. 15–22.

[31] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[32] Y. Song, R. Angel, M. Müller, V. Koltun, and D. Scaramuzza, "Reaching the limit in autonomous racing: Optimal control versus reinforcement learning," *Science Robotics*, 2023.

[33] A. Ren, S. Veer, and A. Majumdar, "Generalization guarantees for imitation learning," in *Proceedings of the 2020 Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, J. Kober, F. Ramos, and C. Tomlin, Eds., vol. 155. PMLR, 16–18 Nov 2021, pp. 1426–1442.

[34] I. Radosavovic, T. Xiao, S. James, P. Abbeel, J. Malik, and T. Darrell, "Real-world robot learning with masked visual pre-training," *CoRL*, 2022.

[35] L. Yen-Chen, A. Zeng, S. Song, P. Isola, and T.-Y. Lin, "Learning to see before learning to act: Visual pre-training for manipulation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 7286–7293.

[36] A. Zhan, R. Zhao, L. Pinto, P. Abbeel, and M. Laskin, "Learning visual robotic control efficiently with contrastive pre-training and data augmentation," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 4040–4047.

[37] 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.

[38] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 15 979–15 988.

[39] M. Shridhar, L. Manuelli, and D. Fox, "Cliport: What and where pathways for robotic manipulation," in *Proceedings of the 5th Conference on Robot Learning (CoRL)*, 2021.

[40] O. Henaff, "Data-efficient image recognition with contrastive predictive coding," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 4182–4192.

[41] S. Qiu, L. Wang, C. Bai, Z. Yang, and Z. Wang, "Contrastive UCB: Provably efficient contrastive self-supervised learning in online reinforcement learning," in *Proceedings of the 39th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Eds., vol. 162. PMLR, 17–23 Jul 2022, pp. 18 168–18 210.

[42] G. Liu, C. Zhang, L. Zhao, T. Qin, J. Zhu, L. Jian, N. Yu, and T.-Y. Liu, "Return-Based Contrastive Representation Learning for Reinforcement Learning," in *Ninth International Conference on Learning Representations(ICLR)*, Jan. 2021.

[43] B. Eysenbach, T. Zhang, S. Levine, and R. R. Salakhutdinov, "Contrastive learning as goal-conditioned reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 35 603–35 620, 2022.

[44] A. Correia and L. A. Alexandre, "Multi-view contrastive learning from demonstrations," in *2022 Sixth IEEE International Conference on Robotic Computing (IRC)*. IEEE, 2022, pp. 338–344.

[45] E. Kaufmann, L. Bauersfeld, and D. Scaramuzza, "A benchmark comparison of learned control policies for agile quadrotor flight," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022.

[46] D. Hanover, A. Loquercio, L. Bauersfeld, A. Romero, R. Penicka, Y. Song, G. Cioffi, E. Kaufmann, and D. Scaramuzza, "Autonomous drone racing: A survey," *arXiv e-prints, pp. arXiv–2301*, 2023.

[47] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm, "Pixel-wise view selection for unstructured multi-view stereo," in *European Conference on Computer Vision (ECCV)*, 2016.

[48] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[49] D. Chen, B. Zhou, V. Koltun, and P. Krähenbühl, "Learning by cheating," in *Proceedings of the Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, L. P. Kaelbling, D. Kragic, and K. Sugiura, Eds., vol. 100. PMLR, 30 Oct–01 Nov 2020, pp. 66–75.

[50] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 627–635.

[51] Y. Song, M. Steinweg, E. Kaufmann, and D. Scaramuzza, "Autonomous drone racing with deep reinforcement learning," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 1205–1212.

[52] F. Torabi, G. Warnell, and P. Stone, "Behavioral cloning from observation," in *IJCAI*, 2018, pp. 4950–4957.

[53] F. Wang and H. Liu, "Understanding the behaviour of contrastive loss," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 2495–2504.

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

[55] Y. Song, S. Naji, E. Kaufmann, A. Loquercio, and D. Scaramuzza, "Flightmare: A flexible quadrotor simulator," in *Conference on Robot Learning*, 2020.

[56] Y. Song, M. Steinweg, E. Kaufmann, and D. Scaramuzza, "Autonomous drone racing with deep reinforcement learning," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 1205–1212.

[57] C. Lea, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks: A unified approach to action segmentation," in *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14*. Springer, 2016, pp. 47–54.

[58] B. O. Community, *Blender - a 3D modelling and rendering package*, Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. [Online]. Available: http://www.blender.org

[59] V. Usenko, N. Demmel, and D. Cremers, "The double sphere camera model," in *2018 International Conference on 3D Vision (3DV)*. IEEE, 2018, pp. 552–560.

[60] 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.

[61] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.

[62] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015.

[63] 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," *arXiv preprint arXiv:2304.07193*, 2023.

[64] L. Bauersfeld, E. Kaufmann, P. Foehn, S. Sun, and D. Scaramuzza, "Neurobem: Hybrid aerodynamic quadrotor model," in *Proceedings of Robotics: Science and Systems*, 2021.

[65] P. Foehn, E. Kaufmann, A. Romero, R. Penicka, S. Sun, L. Bauersfeld, T. Laengle, G. Cioffi, Y. Song, A. Loquercio, and D. Scaramuzza, "Agilicious: Open-source and open-hardware agile quadrotor for vision-based flight," *Science Robotics*, vol. 7, no. 67, 2022.

[66] A. Romero, Y. Song, and D. Scaramuzza, "Actor-critic model predictive control," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024.