2021 I-RIM Conference
October 8-10, Rome, Italy
ISBN: 9788894580525
DOI:10.5281/zenodo.6367974

Abstraction: A Framework for Knowledge Transfer Between Domains

Antonio Loquercio   Davide Scaramuzza

***Abstract*—In this extended abstract, we propose a theoretical framework for knowledge transfer between domains, *e.g.* from simulation to the real world. The framework is based on the concept of input-output abstraction, whose goal is to minimize the distance between domains while keeping enough information to solve the task. The value of this framework is twofold. First, it provides the theoretical background for justifying the empirical finding of prior work that intermediate representations are useful for action. Second, it suggests a practical algorithm for zero-shot simulation to reality transfer. We demonstrate this framework on two challenging tasks: drone racing and high-speed navigation in the wild. A video demonstrating the applications can be found at: https://youtu.be/uTWcC6IBsE4.**

I. Theoretical Derivation

Our approach to bridging the gap between simulation and reality is to leverage *abstraction* [1]. Rather than operating on raw sensory input, we propose to learn a sensorimotor policy operating on intermediate representation produced by a perception module [2]. This intermediate representations is more consistent across simulation and reality than raw visual input, but still has the information to solve the task.

We now formally show that training a network on abstractions of sensory input reduces the gap between simulation and reality. Let $M(\boldsymbol{z} \mid \boldsymbol{s}), L(\boldsymbol{z} \mid \boldsymbol{s})\colon \mathbb{S} \rightarrow \mathbb{O}$ denote the observation models in the real world and in simulation, respectively. Such models describe how a raw sensor measurement $\boldsymbol{z}$ senses a state $\boldsymbol{s}$. We further define $\pi_r = \mathbb{E}_{\boldsymbol{o}_r \sim M(\boldsymbol{s})}[\pi(\boldsymbol{o}_r[k])]$ and $\pi_s = \mathbb{E}_{\boldsymbol{o}_s \sim L(\boldsymbol{s})}[\pi(\boldsymbol{o}_s[k])]$ as the realizations of the policy π in the real world and in simulation. The following lemma shows that, disregarding actuation differences, the distance between the observation models upper-bounds the gap in performance in simulation and reality.

Lemma 1. *For a Lipschitz-continuous policy π the simulation-to-reality gap $J(\pi_r) - J(\pi_s)$ is upper-bounded by*

$$J(\pi_r) - J(\pi_s) \leq C_{\pi_s} K \mathbb{E}_{\rho(\pi_r)}\big[DW(M, L)\big], \tag{1}$$

where K denotes the Lipschitz constant.

Proof. The lemma follows directly from the fact that

$$\begin{aligned} DW(\pi_r, \pi_s) &= \inf_{\gamma \in \Pi(\boldsymbol{o}_r, \boldsymbol{o}_s)} \mathbb{E}_{(\boldsymbol{o}_r, \boldsymbol{o}_s)}[d_p(\pi_r, \pi_s)] \\ &\leq K \inf_{\gamma \in \Pi(\boldsymbol{o}_r, \boldsymbol{o}_s)} \mathbb{E}_{(\boldsymbol{o}_r, \boldsymbol{o}_s)}[d_o(\boldsymbol{o}_r, \boldsymbol{o}_s)] \\ &= K \cdot DW(M, L), \end{aligned}$$

where d_o and d_p are distances in observation and action space, respectively. □

The authors are with the Dep. of Informatics, University of Zurich, and Dep. of Neuroinformatics, the University of Zurich and ETH Zurich, Switzerland (http://rpg.ifi.uzh.ch). This work was supported by the National Centre of Competence in Research (NCCR) Robotics through the Swiss National Science Foundation (SNSF) and the SNSF-ERC Starting Grant.

Fig. 1. We present a general approach to for zero-shot transfer of sensorimotor policies from simulation (left) to real world (right). To account for the large differences in appearance and physics, we propose to abstract the input and output of the policy to minimize the distance between domains.

We now consider the effect of abstraction of the input observations. Let f be a mapping of the observations such that

$$DW(f(M), f(L)) \leq DW(M, L). \tag{2}$$

The mapping f is task-dependent and is generally designed – with domain knowledge – to contain sufficient information to solve the task while being invariant to nuisance factors. In our case, we use feature tracks as an abstraction of camera frames. The feature tracks are provided by a visual-inertial odometry (VIO) system. In contrast to camera frames, feature tracks primarily depend on scene geometry, rather than surface appearance. We also make inertial measurements independent of environmental conditions, such as temperature and pressure, by integration and de-biasing. As such, our input representations fulfill the requirements of Eq. (2).

As the following lemma shows, training on such representations reduces the gap between task performance in simulation and the real world.

Lemma 2. *A policy that acts on an abstract representation of the observations $\pi_f\colon f(\mathbb{O}) \rightarrow \mathbb{U}$ has a lower simulation-to-reality gap than a policy $\pi_o\colon \mathbb{O} \rightarrow \mathbb{U}$ that acts on raw observations.*

Proof. The lemma follows directly from (1) and (2). □

The main question that remains open is how to find such abstraction functions. We propose two ways to do it: either learning them end-to-end with the task or pre-defining them using domain knowledge.

II. Applications

We apply the proposed framework to two tasks: *drone racing*, where the abstraction function is learned jointly with

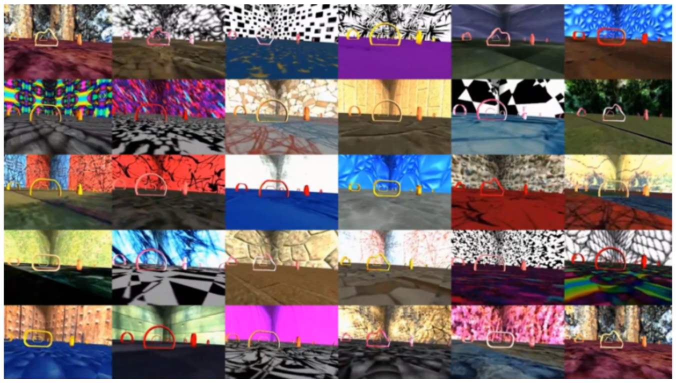

Fig. 2. The perception block of our system, represented by a convolutional neural network (CNN), is trained only with non-photorealistic simulation data. Due to the abundance of such data, generated with domain randomization, the trained CNN can be deployed on a physical quadrotor without any finetuning.

Fig. 3. Autonomous flight in the wild: we train a policy exclusively in simulation by leveraging abstractions of the inputs pre-defined by the user.

the sensorimotor policy, and *high-speed navigation in the wild*, where the abstraction is pre-defined by the user. In both cases, we use a convolutional neural network to compute a receding-horizon trajectory, which we track with a model-predictive controller [3]. Training was done by imitation learning on a specifically designed expert policy with access to privileged information about the state of the drone and of the environment.

For the first demonstrator, we use as expert a model-predictive controller tracking a time-optimal trajectory [4] passing through all the gates. We train the abstraction function together with the policy by using domain randomization [5]. Specifically, we randomize all the features which are unimportant for predictions, *i.e.* illumination, gate shape, floor texture, and background. A sample of the training data generated by this process can be observed in Fig. 2. The abundance of simulated data makes our system more robust than its counterpart trained with real-world data to changes of illumination and gate appearance [5]. However, domain randomization requires strong simulation engineering and expensive trial and error in the real world to define the randomization bounds.

As a second demonstrator, we train a policy to fly a quadrotor at high speeds in a variety of environments with complex obstacle geometry. Similarly to the previous drone racing task, we train the policy *exclusively* in simulation. However, in contrast to the previous demonstrator, we pre-define the input abstraction function to minimize the distance between simulation and real world. Specifically, we utilize a stereo matching algorithm to provide depth images as input to the policy. At training time, disparities are computed on simulated stereo frames. In the physical world, the policy receives as input depth computed from an Intel RealSense D435i. We empirically show that this input representation is rich enough to safely navigate through complex environments and abstract enough to bridge simulation and reality. In addition, by using a stereo matching algorithm on simulated frames, we guarantees a strong similarity of the noise models between simulated and real observations. This gives our policy robustness against common perceptual artifacts in the real depth sensor. A qualitative example of flight in the wild is shown in Figure 3. Instead of learning them end-to-end, pre-defining abstractions favors sample efficiency, simplifies training, and promotes generalization [2]. However, due to human biases, the user-defined abstractions could potentially be suboptimal to the downstream task.

III. Discussion

We have shown the validity of the proposed framework for transfer learning via abstraction on two challenging applications: drone racing to high-speed navigation in unstructured environments. The framework theoretically motivates why input/output abstractions are effective in transferring knowledge between domains. The main limitation of the proposed approach is that it can only account for the differences between domains that can be eliminated by abstractions, e.g. interactions with other (artificial or biological) agents. Such effects might be too complex or computationally intensive to simulate. In these cases, the sensorimotor policies won't be able to generalize zero-shot to the real world. To address this limitation, the policy will need an online adaptation to the environment and task. Doing so in the real world could be challenging due to the lack of privileged information or explicit reward signals but could be supported, for example, by self-supervised learning [6].

References

[1] M. Müller, A. Dosovitskiy, B. Ghanem, and V. Koltun, "Driving policy transfer via modularity and abstraction," in *Conference on Robot Learning (CoRL)*, 2018.

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

[3] D. Falanga, P. Foehn, P. Lu, and D. Scaramuzza, "Pampc: Perception-aware model predictive control for quadrotors," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*. IEEE, 2018, pp. 1–8.

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

[5] A. Loquercio, E. Kaufmann, R. Ranftl, A. Dosovitskiy, V. Koltun, and D. Scaramuzza, "Deep drone racing: From simulation to reality with domain randomization," *IEEE Transactions on Robotics*, vol. 36, no. 1, pp. 1–14, 2019.

[6] N. Hansen, R. Jangir, Y. Sun, G. Alenyà, P. Abbeel, A. A. Efros, L. Pinto, and X. Wang, "Self-supervised policy adaptation during deployment," *arXiv preprint arXiv:2007.04309*, 2020.