Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

b-it Bonn-Aachen International Center for Information Technology

Institute for AI and Autonomous Systems

Sim-to-Real Transfer

Making Simulation-Based Knowledge Useful in Reality

Dr. Alex Mitrevski
Master of Autonomous Systems

Structure

Digital Object Identifier 10.1109/ACCESS.2021.3126658

Crossing the Reality Gap: A Survey on Sim-to-Real Transferability of Robot Controllers in Reinforcement Learning

ERICA SALVATO, GIANFRANCO FENU, ERIC MEDVET, AND FELICE ANDREA PELLEGRINO, (Member, IEEE)

Sim-to-Real Transfer in Deep Reinforcement Learning for Robotics: a Survey

Wenshuai Zhao[1], Jorge Peña Queralta[1], Tomi Westerlund[1]

2019 International Conference on Robotics and Automation (ICRA)
Palais des congres de Montreal, Montreal, Canada, May 20-24, 2019

Quantifying the Reality Gap in Robotic Manipulation Tasks

Jack Collins[1,2], David Howard[2] and Jürgen Leitner[1,3]

- Sim-to-real motivation
- Sim-to-real methods

Sim-to-Real Motivation

What is Sim-to-Real?

J. P. R. Belo and R. A. F. Romero, "A Social Human-Robot Interaction Simulator for Reinforcement Learning Systems," in *Proc. 20th Int. Conf. Advanced Robotics (ICAR)*, 2021, pp. 350–355.

A. H. Qureshi et al., "Robot gains social intelligence through multimodal deep reinforcement learning," in *Proc. IEEE-RAS 16th Int. Conf. Humanoid Robots (Humanoids)*, 2016, pp. 745–751.

- In many learning scenarios in robotics, **collecting real-world for learning data can be impractical or dangerous**

What is Sim-to-Real?

J. P. R. Belo and R. A. F. Romero, "A Social Human-Robot Interaction Simulator for Reinforcement Learning Systems," in *Proc. 20th Int. Conf. Advanced Robotics (ICAR)*, 2021, pp. 350–355.

A. H. Qureshi et al., "Robot gains social intelligence through multimodal deep reinforcement learning," in *Proc. IEEE-RAS 16th Int. Conf. Humanoid Robots (Humanoids)*, 2016, pp. 745–751.

- In many learning scenarios in robotics, **collecting real-world for learning data can be impractical or dangerous**
- For this reason, data collection for learning is often done in a **simulated environment that models the target real-world environment** as closely as possible

What is Sim-to-Real?

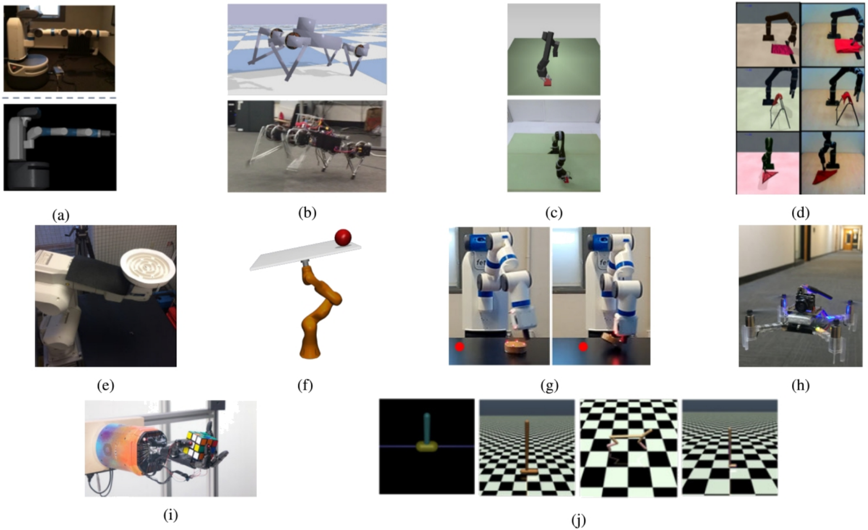
J. P. R. Belo and R. A. F. Romero, "A Social Human-Robot Interaction Simulator for Reinforcement Learning Systems," in *Proc. 20th Int. Conf. Advanced Robotics (ICAR)*, 2021, pp. 350–355.

A. H. Qureshi et al., "Robot gains social intelligence through multimodal deep reinforcement learning," in *Proc. IEEE-RAS 16th Int. Conf. Humanoid Robots (Humanoids)*, 2016, pp. 745–751.

- In many learning scenarios in robotics, **collecting real-world for learning data can be impractical or dangerous**
- For this reason, data collection for learning is often done in a **simulated environment that models the target real-world environment** as closely as possible
- Sim-to-real transfer is the problem of **adapting a simulation-based model to the real world**

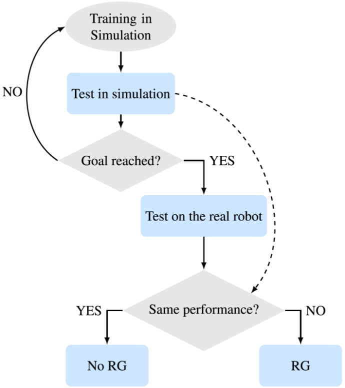
Sim-to-real transfer is the process of making a model that was trained in a simulated environment suitable for use in the real, target environment

Sim-to-Real Applications



Reality Gap

Crossing the Reality Gap: A Survey on Sim-to-Real Transferability of Robot Controllers in Reinforcement Learning
ERICA SALVATO, GIANFRANCO FENU, ERIC MEDVET, AND FELICE ANDREA PELLEGRINO, (Member, IEEE)

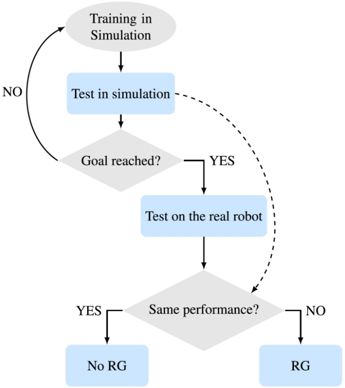


- A simulation is never a completely faithful model, but merely **an approximation of physical entities and phenomena**
  - For instance, object are often modelled as meshes, so object interactions are expressed on the mesh elements
  - Physical phenomena are expressed through (partial) differential equation models

Reality Gap

Crossing the Reality Gap: A Survey on Sim-to-Real Transferability of Robot Controllers in Reinforcement Learning

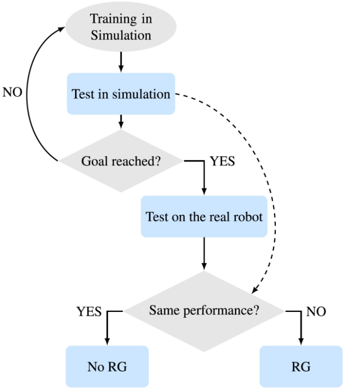
ERICA SALVATO, GIANFRANCO FENU, ERIC MEDVET, AND FELICE ANDREA PELLEGRINO, (Member, IEEE)



- A simulation is never a completely faithful model, but merely **an approximation of physical entities and phenomena**
  - For instance, object are often modelled as meshes, so object interactions are expressed on the mesh elements
  - Physical phenomena are expressed through (partial) differential equation models
- As a result, **a model learned in simulation may not be directly usable in the real environment**

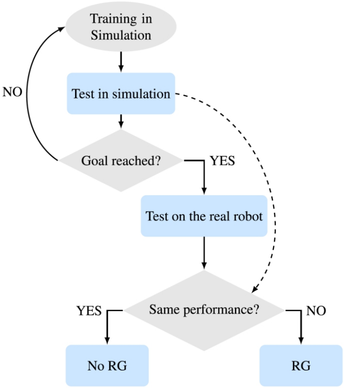
Reality Gap

Crossing the Reality Gap: A Survey on Sim-to-Real Transferability of Robot Controllers in Reinforcement Learning
ERICA SALVATO, GIANFRANCO FENU, ERIC MEDVET, AND FELICE ANDREA PELLEGRINO, (Member, IEEE)



- A simulation is never a completely faithful model, but merely **an approximation of physical entities and phenomena**
  - For instance, object are often modelled as meshes, so object interactions are expressed on the mesh elements
  - Physical phenomena are expressed through (partial) differential equation models
- As a result, **a model learned in simulation may not be directly usable in the real environment**
- The reality gap refers to the **difference in performance when a model learned in simulation is applied in the real world**

Reality Gap



- A simulation is never a completely faithful model, but merely **an approximation of physical entities and phenomena**
  - For instance, object are often modelled as meshes, so object interactions are expressed on the mesh elements
  - Physical phenomena are expressed through (partial) differential equation models
- As a result, **a model learned in simulation may not be directly usable in the real environment**
- The reality gap refers to the **difference in performance when a model learned in simulation is applied in the real world**
- Sim-to-real is thus the problem of **reducing this gap** so that models acquired in simulation can be transferred to the corresponding real system with reasonable accuracy

Sim-to-Real Challenge Cases

Perceptual models

Models trained in simulation (e.g. for object recognition) may overfit on artificially looking objects

Sim-to-Real Challenge Cases

Perceptual models

Models trained in simulation (e.g. for object recognition) may overfit on artificially looking objects

Object interaction

Simulated contacts may be inaccurate for learning accurate interaction models that translate well to real-world interactions

Sim-to-Real Challenge Cases

Perceptual models

Models trained in simulation (e.g. for object recognition) may overfit on artificially looking objects

Object interaction

Simulated contacts may be inaccurate for learning accurate interaction models that translate well to real-world interactions

Policies

Policies learned in simulation can be affected both by perceptual inaccuracies and by inappropriate contact models

Sim-to-Real Challenge Cases

Perceptual models

Models trained in simulation (e.g. for object recognition) may overfit on artificially looking objects

Object interaction

Simulated contacts may be inaccurate for learning accurate interaction models that translate well to real-world interactions

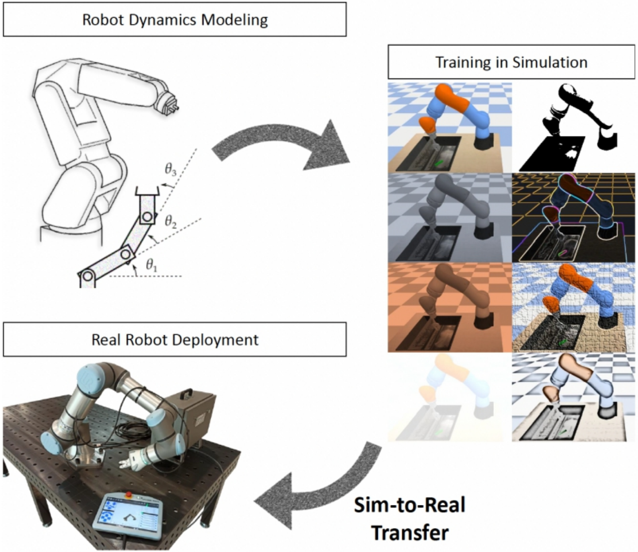
Policies

Policies learned in simulation can be affected both by perceptual inaccuracies and by inappropriate contact models

Human-robot interaction

Simulated human models may not be able to capture important intricacies of how human-robot interaction is performed

Typical Sim-to-Real Workfklow

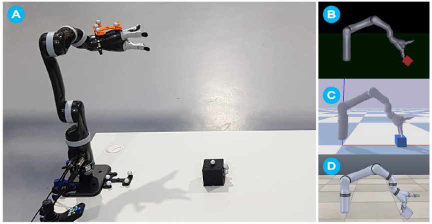


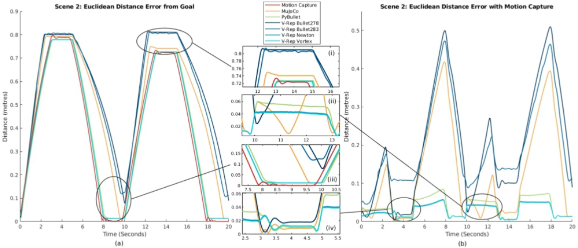
Simulation Errors Without Object Interaction

2019 International Conference on Robotics and Automation (ICRA)
Palais des congres de Montreal, Montreal, Canada, May 20-24, 2019

Quantifying the Reality Gap in Robotic Manipulation Tasks

Jack Collins[1,2], David Howard[2] and Jürgen Leitner[1,3]





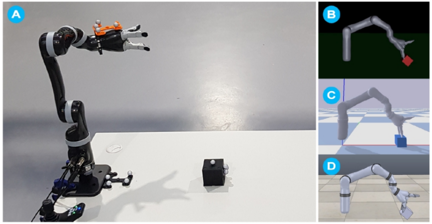
- In an evaluation experiment of multiple commonly used simulations with ground-truth motion capture data, it has been demonstrated that, **with free joint motion, physics engines such as Bullet can produce position errors, but most can follow the arm's motion accurately**

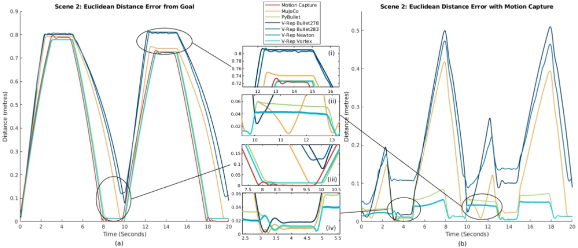
Simulation Errors Without Object Interaction

2019 International Conference on Robotics and Automation (ICRA)
Palais des congres de Montreal, Montreal, Canada, May 20-24, 2019

Quantifying the Reality Gap in Robotic Manipulation Tasks

Jack Collins[1,2], David Howard[2] and Jürgen Leitner[1,3]





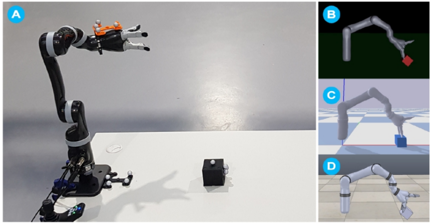
- In an evaluation experiment of multiple commonly used simulations with ground-truth motion capture data, it has been demonstrated that, **with free joint motion, physics engines such as Bullet can produce position errors, but most can follow the arm's motion accurately**
- **Errors are particularly visible when the joint goal orientation changes significantly** — some physics engines are too slow in correcting the error

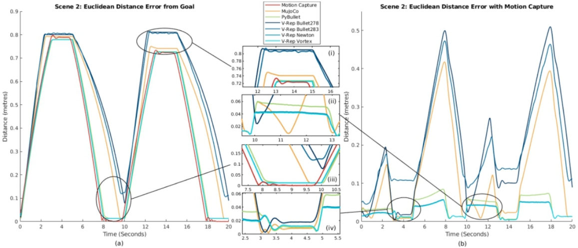
Simulation Errors Without Object Interaction

2019 International Conference on Robotics and Automation (ICRA)
Palais des congres de Montreal, Montreal, Canada, May 20-24, 2019

Quantifying the Reality Gap in Robotic Manipulation Tasks

Jack Collins[1,2], David Howard[2] and Jürgen Leitner[1,3]





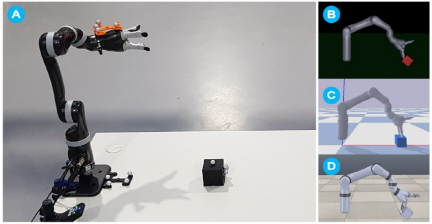
- In an evaluation experiment of multiple commonly used simulations with ground-truth motion capture data, it has been demonstrated that, **with free joint motion, physics engines such as Bullet can produce position errors, but most can follow the arm's motion accurately**
- **Errors are particularly visible when the joint goal orientation changes significantly** — some physics engines are too slow in correcting the error
- From this evaluation, it can also be seen that **the error accumulates visibly for some physics engines**

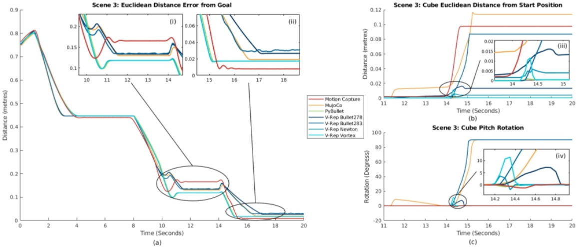
Simulation Errors During Object Interaction

2019 International Conference on Robotics and Automation (ICRA)
Palais des congres de Montreal, Montreal, Canada, May 20-24, 2019

Quantifying the Reality Gap in Robotic Manipulation Tasks

Jack Collins[1,2], David Howard[2] and Jürgen Leitner[1,3]





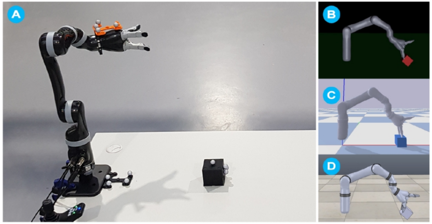
- In the case of interactions with objects (in this case, pushing a simple cube without rotation), **the errors become even more prominent**

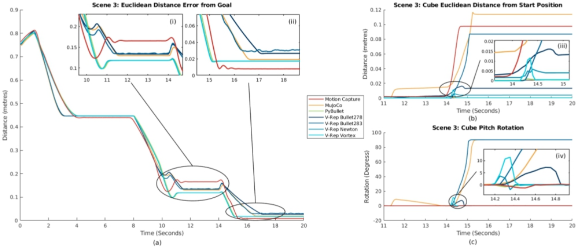
Simulation Errors During Object Interaction

2019 International Conference on Robotics and Automation (ICRA)
Palais des congres de Montreal, Montreal, Canada, May 20-24, 2019

Quantifying the Reality Gap in Robotic Manipulation Tasks

Jack Collins[1,2], David Howard[2] and Jürgen Leitner[1,3]





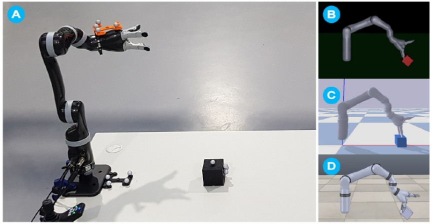
- In the case of interactions with objects (in this case, pushing a simple cube without rotation), **the errors become even more prominent**
- Here, it can be seen that, for some physics engines, such as Vortex or PyBullet, **no or very little interaction with the cube is performed, resulting in a large position error**

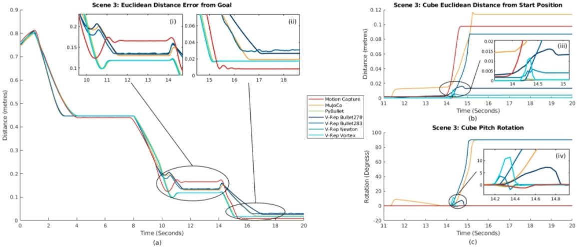
Simulation Errors During Object Interaction

2019 International Conference on Robotics and Automation (ICRA)
Palais des congres de Montreal, Montreal, Canada, May 20-24, 2019

Quantifying the Reality Gap in Robotic Manipulation Tasks

Jack Collins[1,2], David Howard[2] and Jürgen Leitner[1,3]





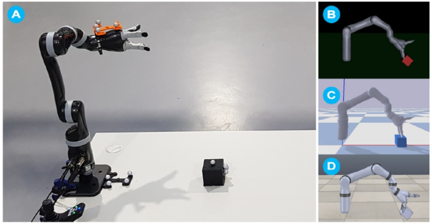
- In the case of interactions with objects (in this case, pushing a simple cube without rotation), **the errors become even more prominent**
- Here, it can be seen that, for some physics engines, such as Vortex or PyBullet, **no or very little interaction with the cube is performed, resulting in a large position error**
- It can also be seen that the cube's position error is large for all engines, which means that **none of the simulators are able to accurately represent the actual cube interaction**

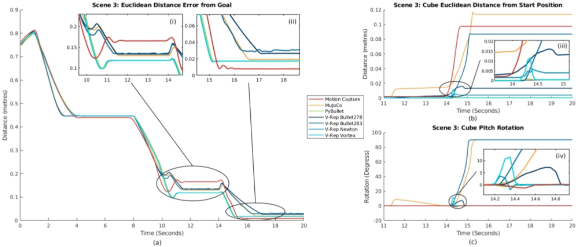
Simulation Errors During Object Interaction

2019 International Conference on Robotics and Automation (ICRA)
Palais des congres de Montreal, Montreal, Canada, May 20-24, 2019

Quantifying the Reality Gap in Robotic Manipulation Tasks

Jack Collins[1,2], David Howard[2] and Jürgen Leitner[1,3]





- In the case of interactions with objects (in this case, pushing a simple cube without rotation), **the errors become even more prominent**
- Here, it can be seen that, for some physics engines, such as Vortex or PyBullet, **no or very little interaction with the cube is performed, resulting in a large position error**
- It can also be seen that the cube's position error is large for all engines, which means that **none of the simulators are able to accurately represent the actual cube interaction**
- This is visible for the rotational motion, where **some engines are shown to lead to a large orientation error due to knocking the cube**

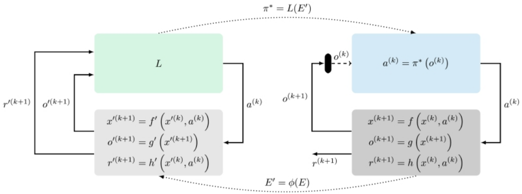
Sim-to-Real in Reinforcement Learning

Digital Object Identifier 10.1109/ACCESS.2021.3126658

Crossing the Reality Gap: A Survey on Sim-to-Real Transferability of Robot Controllers in Reinforcement Learning

ERICA SALVATO, GIANFRANCO FENU, ERIC MEDVET, AND FELICE ANDREA PELLEGRINO, (Member, IEEE)

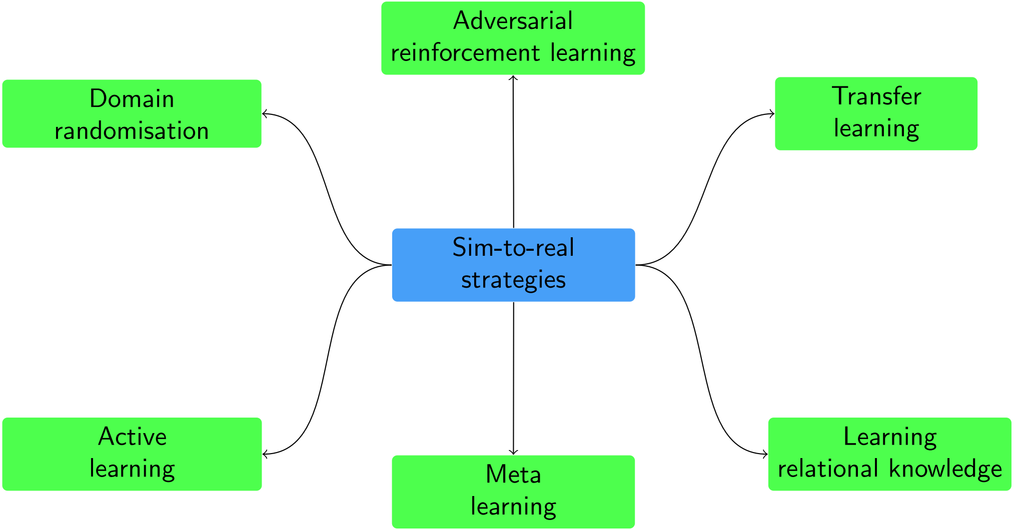
- Sim-to-real is a particularly relevant problem when learning **continuous robotics policies**, namely policies with continuous state and/or action spaces
- Here, the real system is governed by **a transition model** $f : S \times A \to S$, **an observation model** $g : S \to O$, and **a reward function** $h : S \times A \to \mathbb{R}$
- The policy π^* is, however, learned in **a simulated environment** E**, which has an approximate transition model** f', **observation model** g', and **reward model** h'



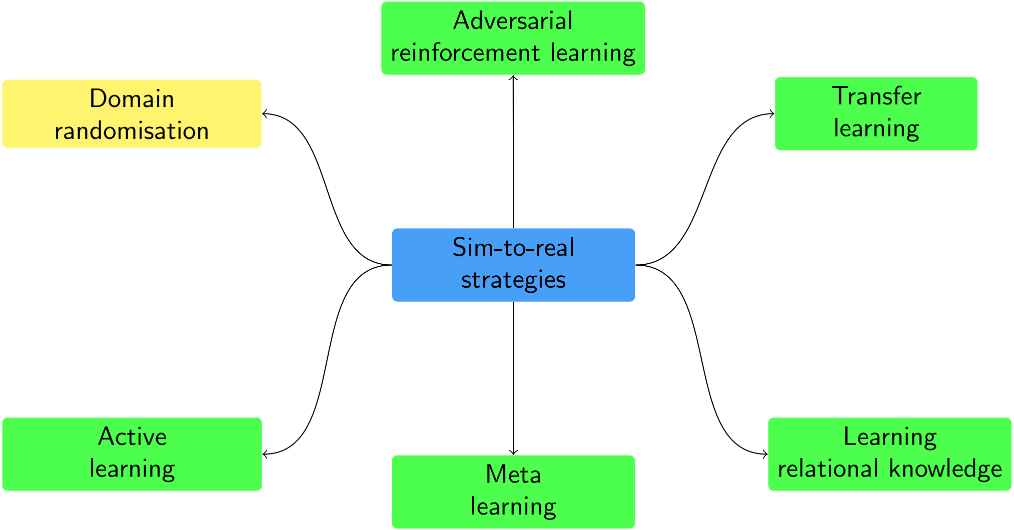
- The transfer problem can be simplified by either **improving the simulated model** (typically difficult) or **designing and learning the policy so that model discrepancies are less detrimental**

Sim-to-Real Methods

Methods for Enabling Sim-to-Real Transfer

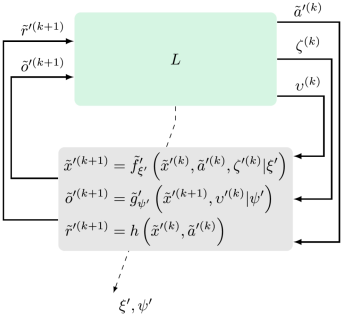


Methods for Enabling Sim-to-Real Transfer



Domain Randomisation

Crossing the Reality Gap: A Survey on Sim-to-Real Transferability of Robot Controllers in Reinforcement Learning
ERICA SALVATO, GIANFRANCO FENU, ERIC MEDVET, AND FELICE ANDREA PELLEGRINO, (Member, IEEE)

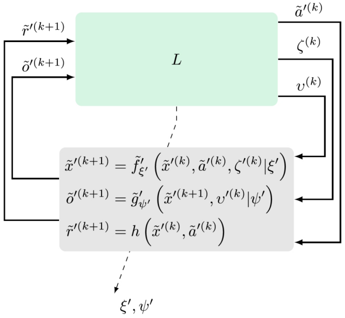


- In domain randomisation, the simulated environment is observed as a **corrupted model** E' with parameters ξ and ψ governing the disturbances of the transition and observation function, respectively

Domain Randomisation

Crossing the Reality Gap: A Survey on Sim-to-Real Transferability of Robot Controllers in Reinforcement Learning

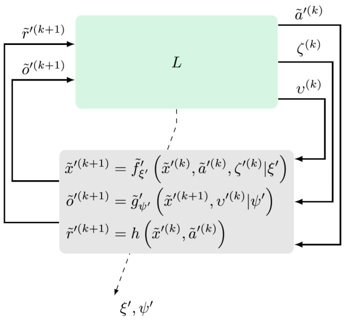
ERICA SALVATO, GIANFRANCO FENU, ERIC MEDVET, AND FELICE ANDREA PELLEGRINO, (Member, IEEE)



- In domain randomisation, the simulated environment is observed as a **corrupted model E'** with parameters ξ and ψ governing the disturbances of the transition and observation function, respectively
- The optimal policy π^* is **trained on a set of similar environments** obtained by varying ξ and ψ

Domain Randomisation

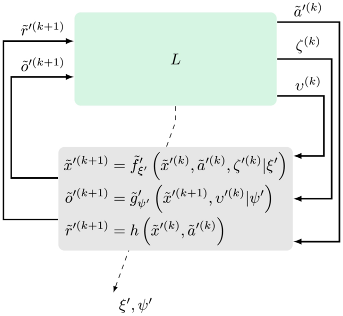
Crossing the Reality Gap: A Survey on Sim-to-Real Transferability of Robot Controllers in Reinforcement Learning
ERICA SALVATO, GIANFRANCO FENU, ERIC MEDVET, AND FELICE ANDREA PELLEGRINO, (Member, IEEE)



- In domain randomisation, the simulated environment is observed as a **corrupted model E'** with parameters ξ and ψ governing the disturbances of the transition and observation function, respectively
- The optimal policy π^* is **trained on a set of similar environments** obtained by varying ξ and ψ
- The objective of this is to **make π^* robust to parameter perturbations** — those will be observed when the policy is transferred to the real world

Domain Randomisation

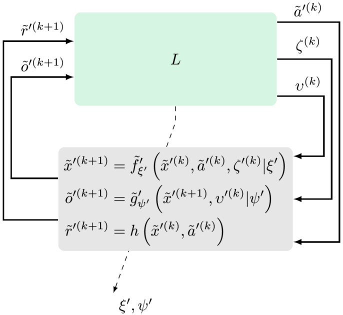
Crossing the Reality Gap: A Survey on Sim-to-Real Transferability of Robot Controllers in Reinforcement Learning
ERICA SALVATO, GIANFRANCO FENU, ERIC MEDVET, AND FELICE ANDREA PELLEGRINO, (Member, IEEE)



- In domain randomisation, the simulated environment is observed as a **corrupted model E'** with parameters ξ and ψ governing the disturbances of the transition and observation function, respectively
- The optimal policy π^* is **trained on a set of similar environments** obtained by varying ξ and ψ
- The objective of this is to **make π^* robust to parameter perturbations** — those will be observed when the policy is transferred to the real world
- It can be useful to randomise a variety of parameters during this process, e.g. physical or camera parameters — but **overdoing randomisation can be detrimental to the learning progress**

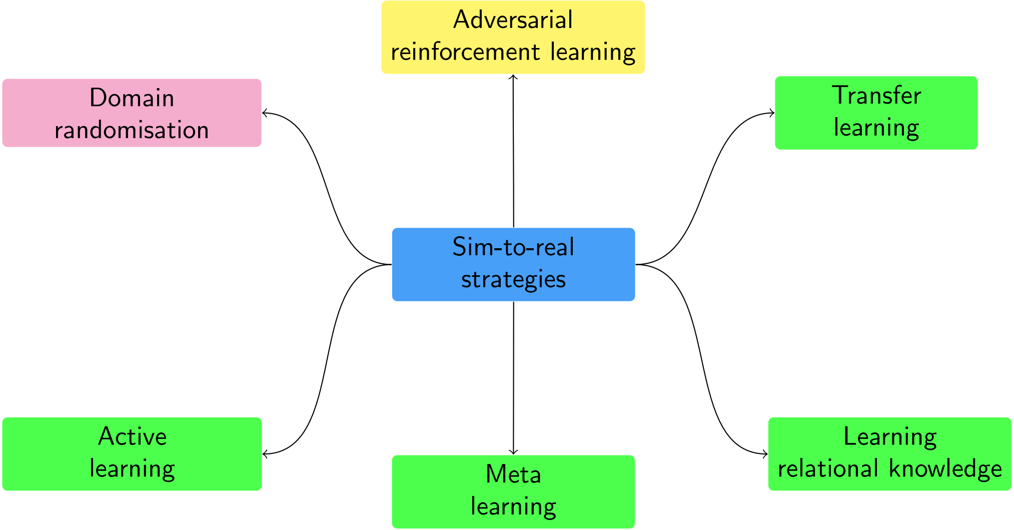
Domain Randomisation

Crossing the Reality Gap: A Survey on Sim-to-Real Transferability of Robot Controllers in Reinforcement Learning
ERICA SALVATO, GIANFRANCO FENU, ERIC MEDVET, AND FELICE ANDREA PELLEGRINO, (Member, IEEE)

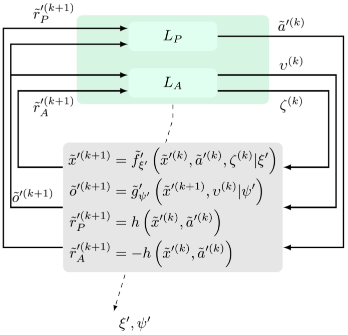


- In domain randomisation, the simulated environment is observed as a **corrupted model E'** with parameters ξ and ψ governing the disturbances of the transition and observation function, respectively
- The optimal policy π^* is **trained on a set of similar environments** obtained by varying ξ and ψ
- The objective of this is to **make π^* robust to parameter perturbations** — those will be observed when the policy is transferred to the real world
- It can be useful to randomise a variety of parameters during this process, e.g. physical or camera parameters — but **overdoing randomisation can be detrimental to the learning progress**
- A randomisation-based learning process may result in a **collection of candidate policies** from which π^* can be selected — for instance, by evaluation over different simulations

Methods for Enabling Sim-to-Real Transfer

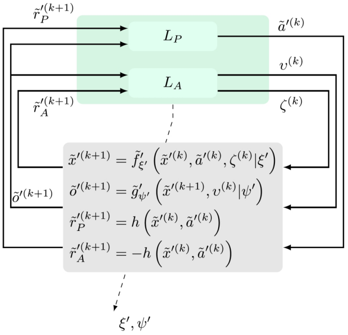


Adversarial Reinforcement Learning



- Adversarial learning also observes the simulated environment E' as a corrupted model of the real environment and **trains a policy that is robust to this corruption**

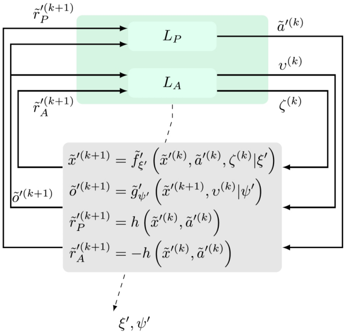
Adversarial Reinforcement Learning



- Adversarial learning also observes the simulated environment E' as a corrupted model of the real environment and **trains a policy that is robust to this corruption**
- As in domain randomisation, **training is done on a collection of environments**, but these are **selected by an adversarial model L_A** that is **trained in parallel with the agent L_P** and which **aims to generate challenging environments** (which lead to a low return)

Adversarial Reinforcement Learning

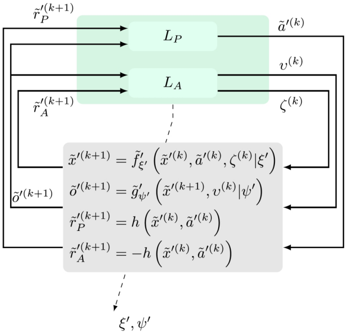
Crossing the Reality Gap: A Survey on Sim-to-Real Transferability of Robot Controllers in Reinforcement Learning
ERICA SALVATO, GIANFRANCO FENU, ERIC MEDVET, AND FELICE ANDREA PELLEGRINO, (Member, IEEE)



- Adversarial learning also observes the simulated environment E' as a corrupted model of the real environment and **trains a policy that is robust to this corruption**
- As in domain randomisation, **training is done on a collection of environments**, but these are **selected by an adversarial model L_A** that is **trained in parallel with the agent L_P** and which **aims to generate challenging environments** (which lead to a low return)
- The policy π_A of L_A is to **select parameters for variying E'**, such that **the policy is trained with the negative of the reward of L_P** (i.e. $\tilde{r}_{L_A} = -\tilde{r}_{L_P}$)

Adversarial Reinforcement Learning

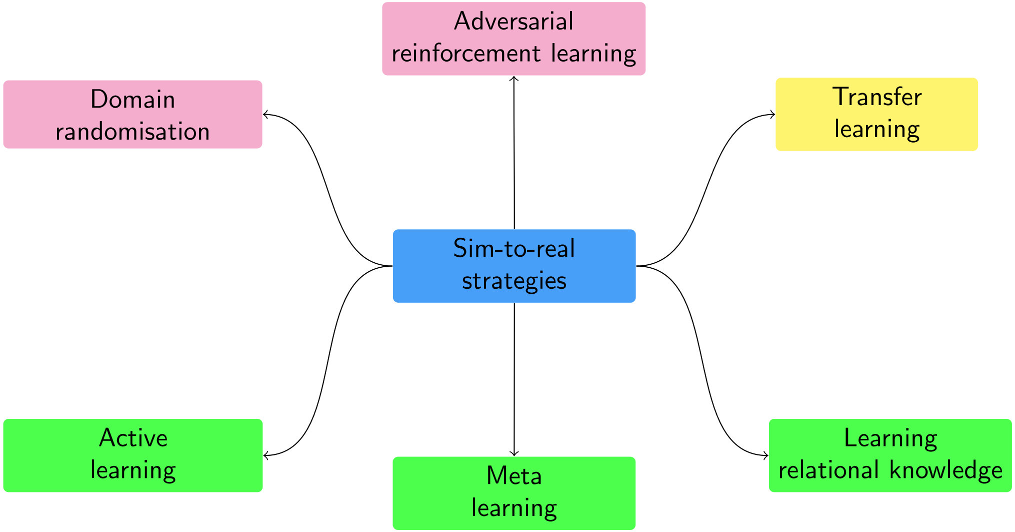
Crossing the Reality Gap: A Survey on Sim-to-Real Transferability of Robot Controllers in Reinforcement Learning
ERICA SALVATO, GIANFRANCO FENU, ERIC MEDVET, AND FELICE ANDREA PELLEGRINO, (Member, IEEE)



- Adversarial learning also observes the simulated environment E' as a corrupted model of the real environment and **trains a policy that is robust to this corruption**
- As in domain randomisation, **training is done on a collection of environments**, but these are **selected by an adversarial model L_A** that is **trained in parallel with the agent L_P** and which **aims to generate challenging environments** (which lead to a low return)
- The policy π_A of L_A is to **select parameters for variying E'**, such that **the policy is trained with the negative of the reward of L_P** (i.e. $\tilde{r}_{L_A} = -\tilde{r}_{L_P}$)
- Denoting the joint return over L_P and L_A as J_{π_P,π_A}, the overall policy learning objective can be expressed as

$$\pi^* = \arg\max_{\pi_P} \arg\min_{\pi_A} J_{\pi_P,\pi_A}$$

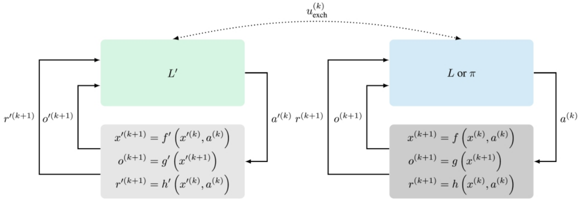
Methods for Enabling Sim-to-Real Transfer



Transfer Learning

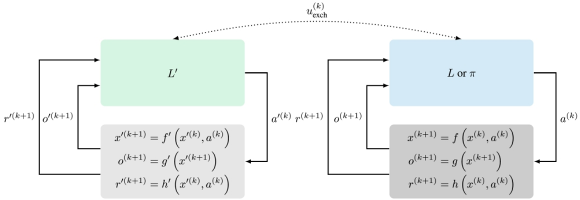
Crossing the Reality Gap: A Survey on Sim-to-Real Transferability of Robot Controllers in Reinforcement Learning

ERICA SALVATO, GIANFRANCO FENU, ERIC MEDVET, AND FELICE ANDREA PELLEGRINO, (Member, IEEE)



- In transfer learning, the policy learned in simulation is not directly applied in the real environment, but is **used as an initialisation for a learning process in the real environment**
  - This initialisation should speed up the physical learning process

Transfer Learning

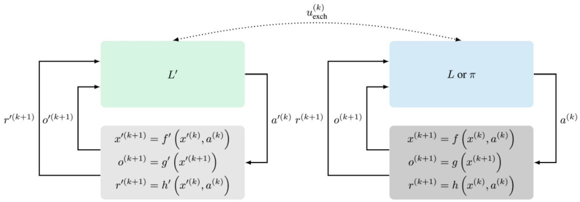


- In transfer learning, the policy learned in simulation is not directly applied in the real environment, but is **used as an initialisation for a learning process in the real environment**
  - This initialisation should speed up the physical learning process
- The transfer can be done in two ways:

Transfer Learning

Crossing the Reality Gap: A Survey on Sim-to-Real Transferability of Robot Controllers in Reinforcement Learning

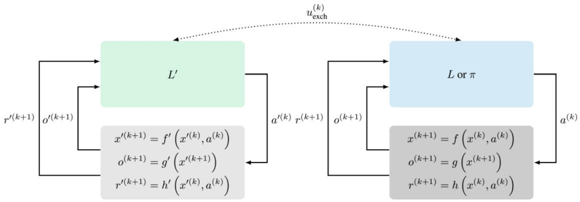
ERICA SALVATO, GIANFRANCO FENU, ERIC MEDVET, AND FELICE ANDREA PELLEGRINO, (Member, IEEE)



- In transfer learning, the policy learned in simulation is not directly applied in the real environment, but is **used as an initialisation for a learning process in the real environment**
  - This initialisation should speed up the physical learning process
- The transfer can be done in two ways:
  - **in a single direction** (the policy learned in E' is used for subsequent real-world training)

Transfer Learning

Crossing the Reality Gap: A Survey on Sim-to-Real Transferability of Robot Controllers in Reinforcement Learning
ERICA SALVATO, GIANFRANCO FENU, ERIC MEDVET, AND FELICE ANDREA PELLEGRINO, (Member, IEEE)

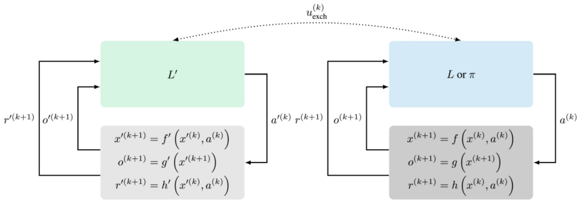


- In transfer learning, the policy learned in simulation is not directly applied in the real environment, but is **used as an initialisation for a learning process in the real environment**
  - This initialisation should speed up the physical learning process
- The transfer can be done in two ways:
  - **in a single direction** (the policy learned in E' is used for subsequent real-world training) or
  - **bidirectionally** (information from the real environment E is also used to improve E')

Transfer Learning

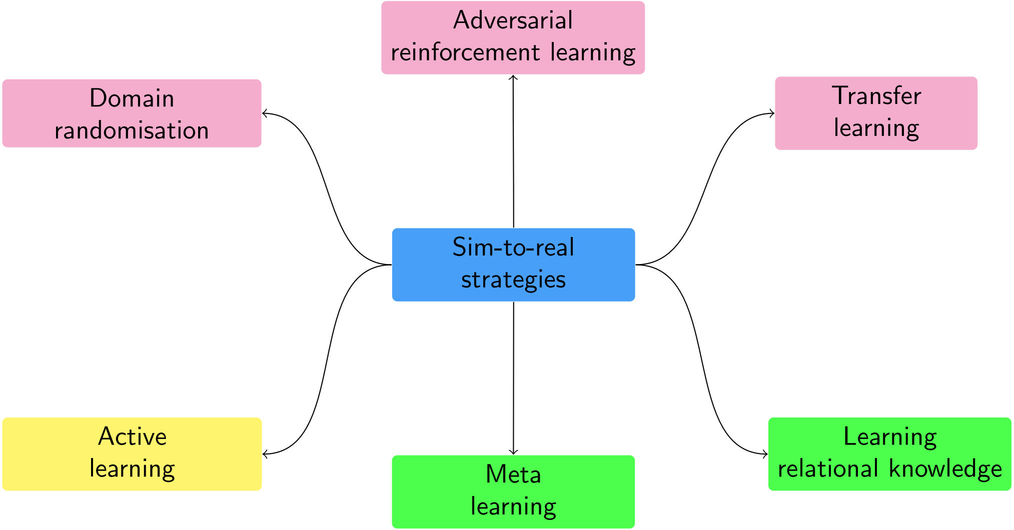
Crossing the Reality Gap: A Survey on Sim-to-Real Transferability of Robot Controllers in Reinforcement Learning

ERICA SALVATO, GIANFRANCO FENU, ERIC MEDVET, AND FELICE ANDREA PELLEGRINO, (Member, IEEE)



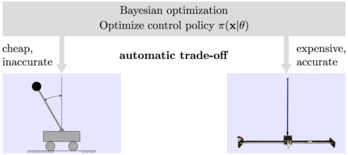
- In transfer learning, the policy learned in simulation is not directly applied in the real environment, but is **used as an initialisation for a learning process in the real environment**
  - This initialisation should speed up the physical learning process
- The transfer can be done in two ways:
  - **in a single direction** (the policy learned in E' is used for subsequent real-world training) or
  - **bidirectionally** (information from the real environment E is also used to improve E')
- The transfer does not need to be done once, but **can be performed iteratively**

Methods for Enabling Sim-to-Real Transfer



Active Learning

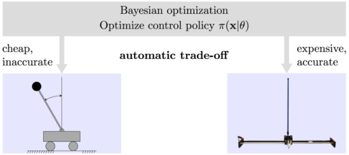
- Another strategy to perform sim-to-real transfer is to **consider both simulated and real experiences during learning**



A. Marco et al., "Virtual vs. real: Trading off simulations and physical experiments in reinforcement learning with Bayesian optimization," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2017, pp. 1557–1563.

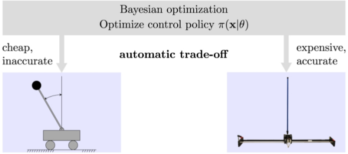
Active Learning

- Another strategy to perform sim-to-real transfer is to **consider both simulated and real experiences during learning**
- **Collecting real experiences is, however, expensive** — that is the motivation for using simulation-based learning in the first place — but active learning can be used for selecting informative experiences



A. Marco et al., "Virtual vs. real: Trading off simulations and physical experiments in reinforcement learning with Bayesian optimization," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2017, pp. 1557–1563.

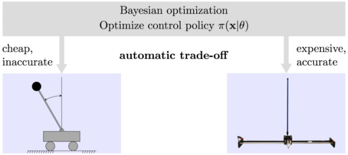
Active Learning



A. Marco et al., "Virtual vs. real: Trading off simulations and physical experiments in reinforcement learning with Bayesian optimization," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2017, pp. 1557–1563.

- Another strategy to perform sim-to-real transfer is to **consider both simulated and real experiences during learning**
- **Collecting real experiences is, however, expensive** — that is the motivation for using simulation-based learning in the first place — but active learning can be used for selecting informative experiences
- Active learning for solving the sim-to-real problem **maintains a cost model $J(\theta)$** based on which

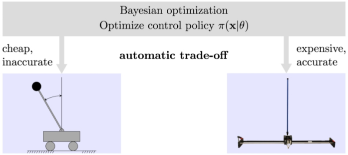
Active Learning



A. Marco et al., "Virtual vs. real: Trading off simulations and physical experiments in reinforcement learning with Bayesian optimization," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2017, pp. 1557–1563.

- Another strategy to perform sim-to-real transfer is to **consider both simulated and real experiences during learning**
- **Collecting real experiences is, however, expensive** — that is the motivation for using simulation-based learning in the first place — but active learning can be used for selecting informative experiences
- Active learning for solving the sim-to-real problem **maintains a cost model $J(\theta)$** based on which
  - **informative experiences are selected for execution** (typically, by reducing the entropy), and

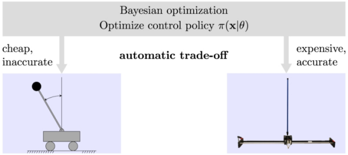
Active Learning



A. Marco et al., "Virtual vs. real: Trading off simulations and physical experiments in reinforcement learning with Bayesian optimization," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2017, pp. 1557–1563.

- Another strategy to perform sim-to-real transfer is to **consider both simulated and real experiences during learning**
- **Collecting real experiences is, however, expensive** — that is the motivation for using simulation-based learning in the first place — but active learning can be used for selecting informative experiences
- Active learning for solving the sim-to-real problem **maintains a cost model $J(\theta)$** based on which
  - **informative experiences are selected for execution** (typically, by reducing the entropy), and
  - a choice is made between **experience collection in simulation or on the real system**

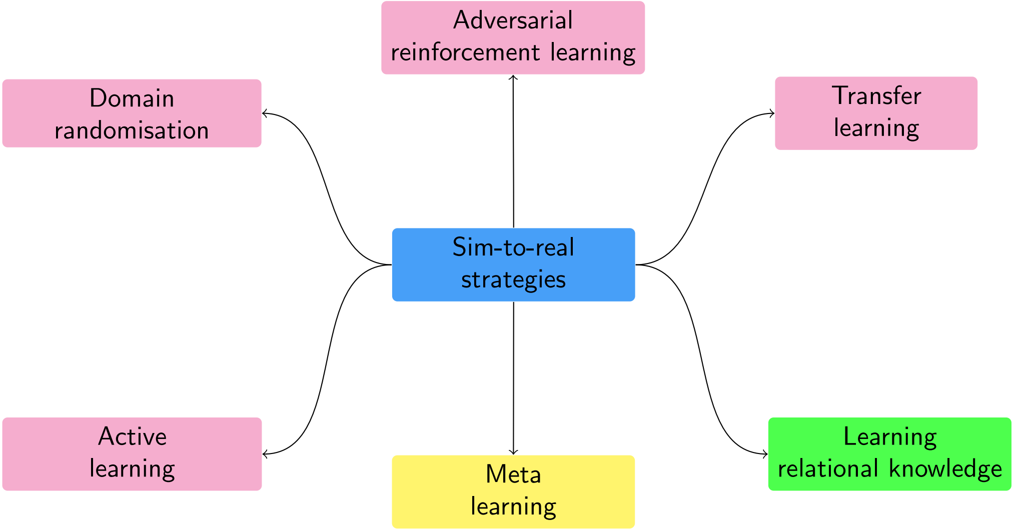
Active Learning



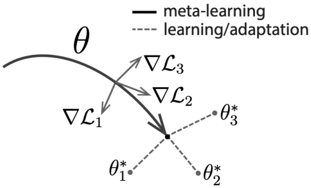
A. Marco et al., "Virtual vs. real: Trading off simulations and physical experiments in reinforcement learning with Bayesian optimization," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2017, pp. 1557–1563.

- Another strategy to perform sim-to-real transfer is to **consider both simulated and real experiences during learning**
- **Collecting real experiences is, however, expensive** — that is the motivation for using simulation-based learning in the first place — but active learning can be used for selecting informative experiences
- Active learning for solving the sim-to-real problem **maintains a cost model $J(\theta)$** based on which
  - **informative experiences are selected for execution** (typically, by reducing the entropy), and
  - a choice is made between **experience collection in simulation or on the real system**
- In Bayesian optimisation, this is achieved by **modelling $J(\theta)$ as a combination** of
  - **the cost estimate in simulation $J_{\text{sim}}(\theta)$** and
  - **an estimate $J_{\text{err}}(\theta)$ of the cost error between the real system and the simulation**

Methods for Enabling Sim-to-Real Transfer



Meta Learning



Algorithm 3 MAML for Reinforcement Learning

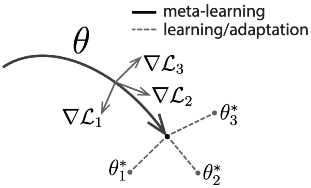
Require: $p(\mathcal{T})$: distribution over tasks
Require: α, β: step size hyperparameters

```
randomly initialize θ
while not done do
    Sample batch of tasks T_i ~ p(T)
    for all T_i do
        Sample K trajectories D = {(x_1, a_1, ...x_H)} using f_θ in T_i
        Evaluate ∇_θ L_{T_i}(f_θ) using D and L_{T_i} in Equation 4
        Compute adapted parameters with gradient descent: θ'_i = θ − α∇_θ L_{T_i}(f_θ)
        Sample trajectories D'_i = {(x_1, a_1, ...x_H)} using f_{θ'_i} in T_i
    end for
    Update θ ← θ − β∇_θ Σ_{T_i~p(T)} L_{T_i}(f_{θ'_i}) using each D'_i and L_{T_i} in Equation 4
end while
```

C. Finn, P. Abbeel, and S. Levine, "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks", in *Proc. 34th Int. Conf. Machine Learning*, vol. 70, 2017, pp. 1126–1135.

- **Meta (reinforcement) learning** can also be used to perform sim-to-real transfer

Meta Learning



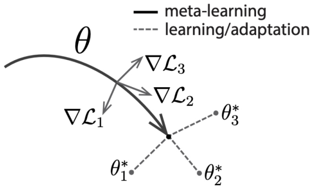
Algorithm 3 MAML for Reinforcement Learning

Require: $p(\mathcal{T})$: distribution over tasks
Require: α, β: step size hyperparameters
randomly initialize θ
while not done **do**
  Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
  for all $\mathcal{T}_i$ **do**
    Sample K trajectories $\mathcal{D} = \{(\mathbf{x}_1, \mathbf{a}_1, \ldots \mathbf{x}_H)\}$ using f_θ in $\mathcal{T}_i$
    Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$ using $\mathcal{D}$ and $\mathcal{L}_{\mathcal{T}_i}$ in Equation 4
    Compute adapted parameters with gradient descent: $\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
    Sample trajectories $\mathcal{D}_i' = \{(\mathbf{x}_1, \mathbf{a}_1, \ldots \mathbf{x}_H)\}$ using $f_{\theta_i'}$ in $\mathcal{T}_i$
  end for
  Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'})$ using each $\mathcal{D}_i'$ and $\mathcal{L}_{\mathcal{T}_i}$ in Equation 4
end while

C. Finn, P. Abbeel, and S. Levine, "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks", in *Proc. 34th Int. Conf. Machine Learning*, vol. 70, 2017, pp. 1126–1135.

- **Meta (reinforcement) learning** can also be used to perform sim-to-real transfer
- The idea behind meta-learning is to **train a model on a distribution of tasks** $p(\mathcal{T})$ from which real tasks can be expected to be sampled

Meta Learning



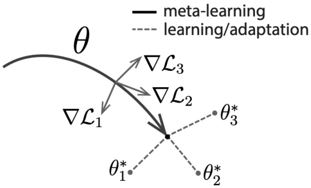
Algorithm 3 MAML for Reinforcement Learning

Require: $p(\mathcal{T})$: distribution over tasks
Require: α, β: step size hyperparameters
randomly initialize θ
while not done **do**
  Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
  for all $\mathcal{T}_i$ **do**
    Sample K trajectories $\mathcal{D} = \{(\mathbf{x}_1, \mathbf{a}_1, ...\mathbf{x}_H)\}$ using f_θ in $\mathcal{T}_i$
    Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$ using $\mathcal{D}$ and $\mathcal{L}_{\mathcal{T}_i}$ in Equation 4
    Compute adapted parameters with gradient descent: $\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
    Sample trajectories $\mathcal{D}_i' = \{(\mathbf{x}_1, \mathbf{a}_1, ...\mathbf{x}_H)\}$ using $f_{\theta_i'}$ in $\mathcal{T}_i$
  end for
  Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'})$ using each $\mathcal{D}_i'$ and $\mathcal{L}_{\mathcal{T}_i}$ in Equation 4
end while

C. Finn, P. Abbeel, and S. Levine, "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks", in *Proc. 34th Int. Conf. Machine Learning*, vol. 70, 2017, pp. 1126–1135.

- **Meta (reinforcement) learning** can also be used to perform sim-to-real transfer
- The idea behind meta-learning is to **train a model on a distribution of tasks** $p(\mathcal{T})$ from which real tasks can be expected to be sampled
- Such a policy can be trained in multiple ways, such as:
  - **by including a memory model** (e.g. using a recurrent neural network)

Meta Learning



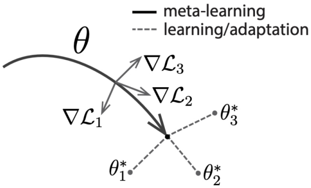
Algorithm 3 MAML for Reinforcement Learning

```
Require: p(T): distribution over tasks
Require: α, β: step size hyperparameters
randomly initialize θ
while not done do
    Sample batch of tasks T_i ~ p(T)
    for all T_i do
        Sample K trajectories D = {(x_1, a_1, ...x_H)} using f_θ in T_i
        Evaluate ∇_θ L_{T_i}(f_θ) using D and L_{T_i} in Equation 4
        Compute adapted parameters with gradient descent: θ'_i = θ − α∇_θ L_{T_i}(f_θ)
        Sample trajectories D'_i = {(x_1, a_1, ...x_H)} using f_{θ'_i} in T_i
    end for
    Update θ ← θ − β∇_θ Σ_{T_i~p(T)} L_{T_i}(f_{θ'_i}) using each D'_i and L_{T_i} in Equation 4
end while
```

C. Finn, P. Abbeel, and S. Levine, "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks", in *Proc. 34th Int. Conf. Machine Learning*, vol. 70, 2017, pp. 1126–1135.

- **Meta (reinforcement) learning** can also be used to perform sim-to-real transfer
- The idea behind meta-learning is to **train a model on a distribution of tasks** $p(\mathcal{T})$ from which real tasks can be expected to be sampled
- Such a policy can be trained in multiple ways, such as:
  - **by including a memory model** (e.g. using a recurrent neural network) or
  - **by enabling policy parameter updates to incorporate data from new tasks** (e.g. using the model-agnostic meta learning (MAML) method)

Meta Learning



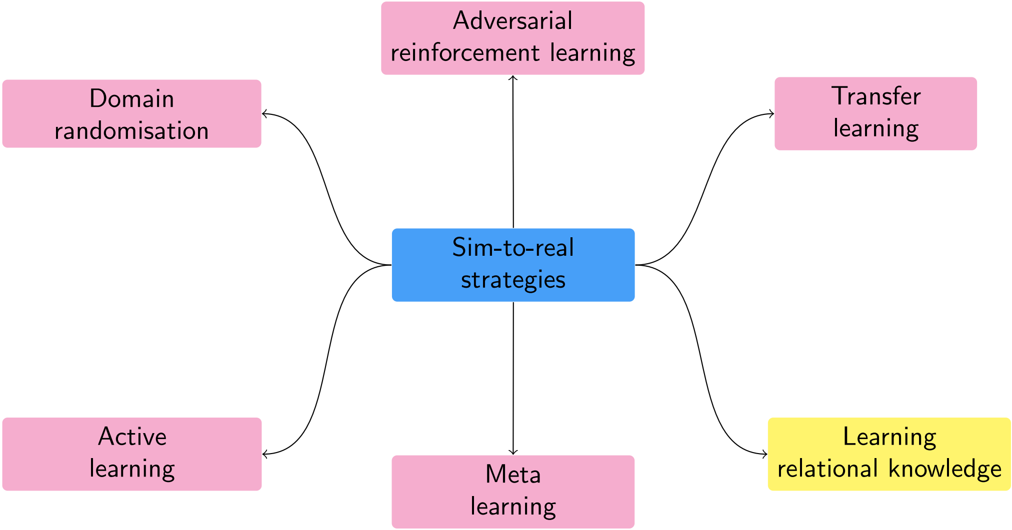
Algorithm 3 MAML for Reinforcement Learning

Require: $p(\mathcal{T})$: distribution over tasks
Require: α, β: step size hyperparameters
randomly initialize θ
while not done **do**
  Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
  for all $\mathcal{T}_i$ **do**
    Sample K trajectories $\mathcal{D} = \{(\mathbf{x}_1, \mathbf{a}_1, ...\mathbf{x}_H)\}$ using f_θ in $\mathcal{T}_i$
    Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$ using $\mathcal{D}$ and $\mathcal{L}_{\mathcal{T}_i}$ in Equation 4
    Compute adapted parameters with gradient descent: $\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
    Sample trajectories $\mathcal{D}_i' = \{(\mathbf{x}_1, \mathbf{a}_1, ...\mathbf{x}_H)\}$ using $f_{\theta_i'}$ in $\mathcal{T}_i$
  end for
  Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'})$ using each $\mathcal{D}_i'$ and $\mathcal{L}_{\mathcal{T}_i}$ in Equation 4
end while

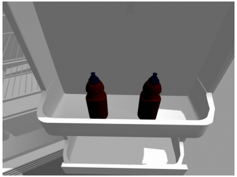
C. Finn, P. Abbeel, and S. Levine, "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks", in *Proc. 34th Int. Conf. Machine Learning*, vol. 70, 2017, pp. 1126–1135.

- **Meta (reinforcement) learning** can also be used to perform sim-to-real transfer
- The idea behind meta-learning is to **train a model on a distribution of tasks** $p(\mathcal{T})$ from which real tasks can be expected to be sampled
- Such a policy can be trained in multiple ways, such as:
  - **by including a memory model** (e.g. using a recurrent neural network) or
  - **by enabling policy parameter updates to incorporate data from new tasks** (e.g. using the model-agnostic meta learning (MAML) method)
- Meta-learning is conceptually related to transfer learning, but **uses a different (meta-)learning objective** that aims to optimise the hyperparameters of the learning process

Methods for Enabling Sim-to-Real Transfer



Learning Relational Knowledge



↓

A. Mitrevski et al., "Improving the reliability of service robots in the presence of external faults by learning action execution models," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2017, pp. 4256–4263.

- A sometimes forgotten sim-to-real strategy is that of **acquiring conceptual knowledge that is more robust to discrepancies** between the simulated and the real environment — for instance, in the form of relational knowledge

Learning Relational Knowledge

A. Mitrevski et al., "Improving the reliability of service robots in the presence of external faults by learning action execution models," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2017, pp. 4256–4263.

- A sometimes forgotten sim-to-real strategy is that of **acquiring conceptual knowledge that is more robust to discrepancies** between the simulated and the real environment — for instance, in the form of relational knowledge
- Relational knowledge can concretely be used to
  - **abstract away brittle details about the environment** (which may otherwise be incorporated into learned policies)

Learning Relational Knowledge

A. Mitrevski et al., "Improving the reliability of service robots in the presence of external faults by learning action execution models," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2017, pp. 4256–4263.

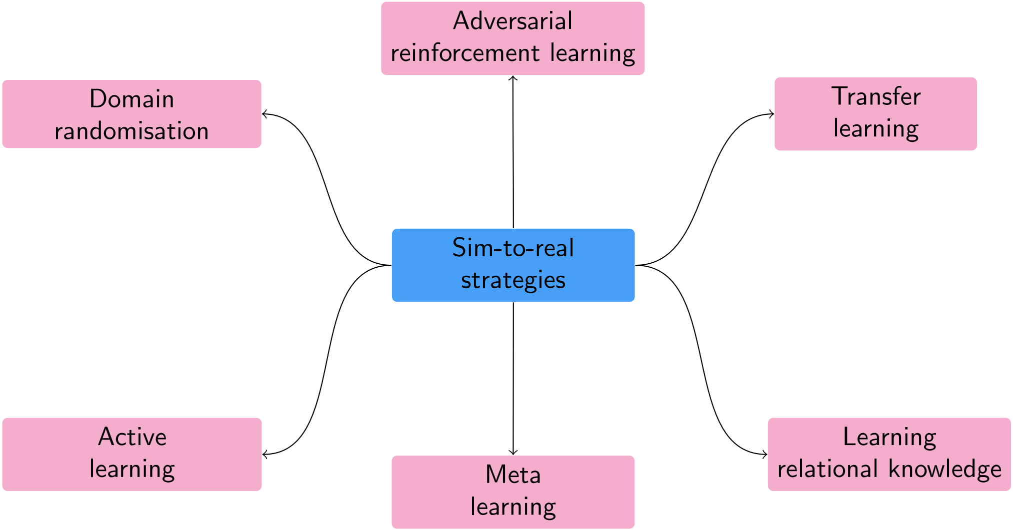
- A sometimes forgotten sim-to-real strategy is that of **acquiring conceptual knowledge that is more robust to discrepancies** between the simulated and the real environment — for instance, in the form of relational knowledge
- Relational knowledge can concretely be used to
  - **abstract away brittle details about the environment** (which may otherwise be incorporated into learned policies) and instead
  - **represent information about invariances that should not change between the simulation and the real world** (e.g. how objects should be positioned with respect to one another so that a task is successfully completed)

Learning Relational Knowledge

A. Mitrevski et al., "Improving the reliability of service robots in the presence of external faults by learning action execution models," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2017, pp. 4256–4263.

- A sometimes forgotten sim-to-real strategy is that of **acquiring conceptual knowledge that is more robust to discrepancies** between the simulated and the real environment — for instance, in the form of relational knowledge
- Relational knowledge can concretely be used to
  - **abstract away brittle details about the environment** (which may otherwise be incorporated into learned policies) and instead
  - **represent information about invariances that should not change between the simulation and the real world** (e.g. how objects should be positioned with respect to one another so that a task is successfully completed)
- Learning relational knowledge does, however, require:
  - **a definition of symbols to learn** or **a procedure to extract new symbols**
  - **explicit state estimation** for estimating the values of the symbols

Methods for Enabling Sim-to-Real Transfer



Methods for Enabling Sim-to-Real Transfer

Sim-to-real strategies

- Adversarial reinforcement learning
- Domain randomisation
- Transfer learning
- Active learning
- Meta learning
- Learning relational knowledge

All of these have their own merits and limitations

Summary

- Sim-to-real transfer refers to the problem of using knowledge that was acquired in simulation in the corresponding real-world environment
- The problem exists because of the reality gap, namely the inevitable discrepancy between simulated and real environments, which stems from the fact that simulations are simplified models of real environments
- There is a variety of methods that can be used for performing sim-to-real transfer — we particularly looked at domain randomisation, adversarial learning, transfer learning, active learning, meta-learning, as well as learning relational knowledge
- There is no clear winner among the existing sim-to-real methods in terms of quality and applicability
    - Which method is best to apply can depend on what is and is not known about the real environment