VR Goggles for Robots: Real-to-sim Domain Adaptation for Visual Control

Jingwei Zhang[*1], **Lei Tai**[*2], **Yufeng Xiong**[1], **Ming Liu**[2], **Joschka Boedecker**[1], **Wolfram Burgard**[1]
[1] University of Freiburg, [2] The Hong Kong University of Science and Technology

Abstract

This paper deals with the *reality gap* from a novel perspective, targeting transferring Deep Reinforcement Learning (DRL) policies learned in simulated environments to the real-world domain for visual control tasks. Instead of adopting the common solutions to the problem by increasing the visual fidelity of synthetic images output from simulators during the training phase, this paper seeks to tackle the problem by translating the real-world image streams back to the synthetic domain during the deployment phase, to *make the robot feel at home*. We propose this as a lightweight, flexible, and efficient solution for visual control, as 1) no extra transfer steps are required during the expensive training of DRL agents in simulation; 2) the trained DRL agents will not be constrained to being deployable in only one specific real-world environment; 3) the policy training and the transfer operations are decoupled, and can be conducted in parallel. Besides this, we propose a conceptually simple yet very effective *shift loss* to constrain the consistency between subsequent frames, eliminating the need for optical flow. We validate the *shift loss* for *artistic style transfer for videos* and *domain adaptation*, and validate our visual control approach in real-world robot experiments. A video of our results is available at: `https://goo.gl/b1xz1s`.

1 Introduction

Pioneered by the Deep Q-network [Mnih *et al.*, 2015] and followed up by various extensions and advancements [Mnih *et al.*, 2016; Lillicrap *et al.*, 2015; Schulman *et al.*, 2015; Schulman *et al.*, 2017], Deep Reinforcement Learning (DRL) algorithms show great potential in solving high-dimensional real-world robotics sensory control tasks. However, DRL methods typically require several millions of training samples, making them infeasible to train directly on real robotic systems. As a result, DRL algorithms are generally trained in simulated environments, then transferred to and deployed in real scenes. However, the *reality gap*, also referred to as the *domain shift*, namely the noise pattern, texture, lighting condition discrepancies, etc., between synthetic renderings and real sensory readings, imposes major challenges for generalizing the sensory control policies trained in simulation to reality.

In this paper, we focus on visual control tasks, where autonomous agents perceive the environment with their onboard cameras, and execute commands based on color image reading streams. A natural way and also the typical choice in the recent literature of dealing with the *reality gap* for visual control, is by increasing the visual fidelity of the simulated images [Bousmalis *et al.*, 2017], by matching the distribution of synthetic images to that of the real ones [Tobin *et al.*, 2017], and by gradually adapting the learned features and representations from the simulated domain to the real-world domain [Rusu *et al.*, 2017]. These *sim-to-real* methods, however, inevitably have to add preprocessing steps for each individual training frame to the already expensive learning pipeline of DRL control policies; also, the complete policy training phase has to be conducted again for each visually different real-world scene. Attempts have also been made in computer graphics to directly increase the quality of the simulators, to make the synthetically rendered images more visually realistic; however, the rendering for detailed and realistic texture and modality often adds to the computational burden.

This paper attempts to tackle the *reality gap* in the visual control domain from a novel perspective, with the aim of adding minimal extra computational burden to the learning pipeline. We cope with the *reality gap* only during the actual deployment phase of agents in real-world scenarios, by adapting the real camera streams to the synthetic modality, so as to translate the unfamiliar or unseen features of real images back into the simulated style, which the agents have already learned how to deal with during training in the simulators.

Compared to other *sim-to-real* methods bridging the *reality gap*, our proposed *real-to-sim* approach, which we refer to as *VR Goggles*, has several appealing properties:

- First of all, our approach is highly lightweight. It does not add any extra processing burden to the training phase of DRL policies.
- Secondly, our proposed method is highly flexible and efficient. Since we decouple the policy training and the adaptation operations, the preparations for transferring

[*]indicates equal contribution.

the polices from simulation to the real world can be conducted in parallel with the training of DRL control policies. From each visually different real-world environment that we expect to deploy the agent in, we just need to collect several (typically on the order of 2000) images, and train a model of *VR Goggles* for each of them. More importantly, we do not need to retrain or finetune the visual control policy for new environments.

As an additional contribution, we propose a new *shift loss*, which enables us to generate consistent synthetic image streams without information to impose temporal constraints, or even sequential training data. We show that *shift loss* is a promising and much cheaper alternative to the constraints imposed by optical flow, and we demonstrate its effectiveness in *artistic style transfer for videos* and *domain adaptation*.

2 Related Works

2.1 *Domain Adaptation*

Domain adaptation, also referred to as *image-to-image translation*, targets translating images from a source domain into a target domain. We here focus on the most general unsupervised methods that require the least manual effort and are applicable to robotics control tasks.

CycleGAN [Zhu *et al.*, 2017a] introduced a cycle-consistent loss to enforce an inverse mapping from the target domain to the source domain on top of the source to target mapping. It does not require paired data from the two domains of interest, and shows convincing results for relatively simple data distributions containing few semantic types. However, in terms of translating between more complex data distributions containing many more semantic types, its results are not as satisfactory, in that permutations of semantics often occur. *CyCADA* [Hoffman *et al.*, 2017] added a semantic constraint on top, to enforce a match between the semantic map of the translated image and that of the input. However, the semantic loss was not added in its experiments on large datasets due to memory limitations.

Following the observation that several most recent and advanced robotics simulators do provide semantic ground truth, and the semantic segmentation literature is quite mature (e.g., [Chen *et al.*, 2017]), we adopt the semantic constraint from *CyCADA* into our method. We are able to include the semantic loss calculation with special configurations (Sec. 4.2).

2.2 *Domain Adaptation* for DRL

DRL approaches have been adopted into robotics control tasks such as manipulation and navigation. Below we review the recent literature with an emphasis on works taking the *reality gap* into consideration.

For manipulation, [Bousmalis *et al.*, 2017] bridged the *reality gap* by adapting synthetic images to the realistic domain during the training phase. However, this addition of an adaptation step before every training iteration can greatly slow down the whole learning pipeline. [Tobin *et al.*, 2017] proposed to ramdomise the texture of objects, lighting conditions, and camera positions during training, in the hope that the learned model will generalize naturally to real-world scenarios. However, such randomization unfortunately cannot be satisfied at a low cost by most of the popular robotic simulators. Moreover, there is no guarantee that these randomized simulations can cover the visual modality of a random real-world scene. [Rusu *et al.*, 2017] deals with the *reality gap* by progressively adapting the learned features and representations of a model trained in simulation to that of the realistic domain. This method, however, still needs to go through an expensive control policy training phase for each visually different real-world scenario.

For navigation, where autonomous agents are expected to encounter sensor readings of environments at a much larger scale than manipulation, the *reality gap* has not been directly dealt with in the literature of learning-based visual control to the best of our knowledge. Some works, however, chose special setups to circumvent the *reality gap*. For example, $2D$ Lidar [Tai *et al.*, 2017; Zhang *et al.*, 2017b] and depth images [Zhang *et al.*, 2017a; Tai *et al.*, 2018] are sometimes chosen as the sensor modality for transferring the navigation policies to the real world, since the discrepancies between the simulated domain and the real-world domain for them are smaller than those for color images. [Zhu *et al.*, 2017b] conducted real-world experiments with visual inputs. But in their setups, the real-world scene is highly visually similar to their simulated environment, which is a condition that can rarely be met in practice.

In this paper, we mainly consider *domain adaptation* for visual navigation tasks using DRL, which has not yet been considered in the literature. We believe the adaptation for navigation is much more challenging than for manipulation, since navigation agents usually work in environments at much larger scales with more complexities than the confined workspace for manipulators. We believe our proposed *real-to-sim* method can be naturally adopted in manipulation.

An important aspect of *domain adaptation*, within the context of dealing with the *reality gap* for DRL, is the consistency between subsequent frames, which has not yet been considered in any of the aforementioned adaptation methods. As a method for solving sequential decision making, the consistency between the subsequent input frames for DRL agents can be critical for the successful fulfillment of their final goals. Apart from the solutions for solving the *reality gap* for DRL, the general *domain adaptation* literature also lacks works considering sequential frames instead of single frames.

Therefore, we look to borrow techniques from other research fields that successfully extend single-frame algorithms to the video domain, among which the most applicable methods are those from the *artistic style transfer* literature.

2.3 *Artistic Style Transfer* for Videos

Artistic style transfer is a technique for transfering the artistic style of artworks to photographs [Johnson *et al.*, 2016].

Artistic style transfer for videos works on video sequences instead of individual frames. It targets generating temporally consistent stylizations for sequential frames. [Ruder *et al.*, 2017] provides a key observation that: a trained stylization network with a total downsampling factor of $\mathbf{s}$ (e.g., $\mathbf{s} = 4$ for a network with 2 convolutional layers of stride 2), is shift invariant to shifts equal to the multiples of $\mathbf{s}$ pixels, but can output significantly different stylizations otherwise. This un-

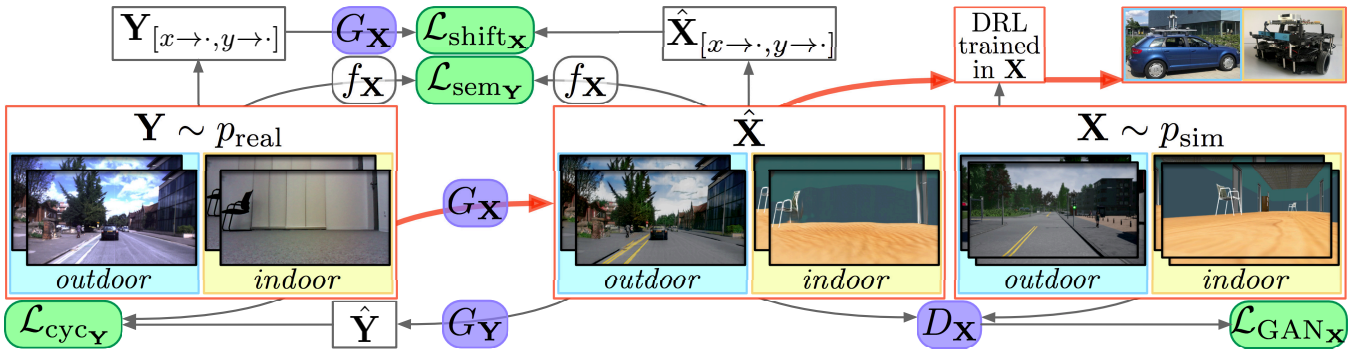

Figure 1: The *VR Goggles* pipeline. We depict the computation of the losses $\mathcal{L}_{\text{GAN}_\mathbf{X}}$, $\mathcal{L}_{\text{cyc}_\mathbf{Y}}$, $\mathcal{L}_{\text{sem}_\mathbf{Y}}$ and $\mathcal{L}_{\text{shift}_\mathbf{X}}$. We show both *outdoor* and *indoor* scenarios for demonstration, where the adaptation for the *outdoor* scene is trained with the semantic loss $\mathcal{L}_{\text{sem}}$ (since its simulated domain *CARLA* has ground truth semantic labels to train a segmentation network $f_\mathbf{X}$), and the *indoor* one without (since its simulated domain *Gazebo* does not provide semantic ground truth). The components marked in *red* are those involved in the final deployment phase: a real sensor reading is captured ($\mathbf{y} \sim p_{\text{real}}$), then passed through the *Goggles* module (generator $G_\mathbf{X}$) to be translated into the simulated domain $\mathbf{X}$, where the DRL agents were originally trained; the translated image $\hat{\mathbf{x}}$ is then fed to DRL agents, to output control commands for autonomous vehicles. For clarity, we skip the counterpart losses $\mathcal{L}_{\text{GAN}_\mathbf{Y}}$, $\mathcal{L}_{\text{cyc}_\mathbf{X}}$, $\mathcal{L}_{\text{sem}_X}$ and $\mathcal{L}_{\text{shift}_Y}$.

desired property (of not being shift invariant) causes the output of the trained network to change significantly for even very small changes in the input, which leads to temporal inconsistency (under the assumption that only relatively limited changes would appear in subsequent frames of the incoming sequential data). However, their solution of adding temporal constraints between generated subsequent frames, is rather expensive, as it requires optical flow as input during deployment. [Huang *et al.*, 2017] incorporated the temporal constraint into the single-frame *artistic style transfer* pipeline and is a relatively cheap solution. However, we believe that constraining optical flow on single input images is not well-defined. We suspect that the improved temporal consistency in [Huang *et al.*, 2017] is actually due to the inexplicitly imposed consistency constraints for regional shifts by optical flow. We validate this suspicion in our experiments (Sec. 4.1).

We believe that the fundamental problem causing the inconsistency (the shift variance) can be solved by an additional constraint of *shift loss*, which we will introduce in Sec. 3.4. We show that the *shift loss* enables us to constrain the consistency between generated subsequent frames, without the need for the relatively expensive optical flow constraint. We argue that for a network that has been properly trained to learn a smooth function approximation, small changes in the input should also result in small changes in the output.

3 Methods

3.1 Problem formulation

We consider visual data sources from two domains: $\mathbf{X}$, containing sequential frames $\{\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \cdots\}$ (e.g., synthetic images output from a simulator; $\mathbf{x} \sim p_{\text{sim}}$, where p_{sim} denotes the simulated data distribution), and $\mathbf{Y}$, containing sequential frames $\{\mathbf{y}_0, \mathbf{y}_1, \mathbf{y}_2, \cdots\}$ (e.g., real camera readings from the onboard camera of a mobile robot; $\mathbf{y} \sim p_{\text{real}}$, where p_{real} denotes the distrbution of the real sensory readings). We emphasize that, although we require our method to generate consistent outputs for sequential inputs, we do not need the training data to be sequential; we formalize it in this way only because some baseline methods have this requirement.

As we have discussed, DRL agents are typically trained in the simulated domain $\mathbf{X}$, while they are expected to perform tasks in the real-world domain $\mathbf{Y}$. And as we have discussed before, we choose to tackle this problem by translating the images from real domain images to the synthetic domain during deployment. In the following we introduce the details of our approach for performing *domain adaptation*. Also to cope with the sequential nature of the incoming data streams, we introduce a technique for constraining the consistency of the translated subsequent frames.

3.2 *CycleGAN* Loss

To achieve this, we first build on top of *CycleGAN* [Zhu *et al.*, 2017a], which learns two generative models to map between domains: $G_\mathbf{Y} : \mathbf{X} \to \mathbf{Y}$, with its discriminator $D_\mathbf{Y}$, and $G_\mathbf{X} : \mathbf{Y} \to \mathbf{X}$, with its discriminator $D_\mathbf{X}$, via training two GANs simultaneously:

$$\begin{aligned}\mathcal{L}_{\text{GAN}_\mathbf{Y}}(G_\mathbf{Y}, D_\mathbf{Y}; \mathbf{X}, \mathbf{Y}) &= \mathbb{E}_\mathbf{y}\left[\log D_\mathbf{Y}(\mathbf{y})\right] \\ &\quad + \mathbb{E}_\mathbf{x}\left[\log(1 - D_\mathbf{Y}(G_\mathbf{Y}(\mathbf{x})))\right], \end{aligned} \tag{1}$$

$$\begin{aligned}\mathcal{L}_{\text{GAN}_\mathbf{X}}(G_\mathbf{X}, D_\mathbf{X}; \mathbf{Y}, \mathbf{X}) &= \mathbb{E}_\mathbf{x}\left[\log D_\mathbf{X}(\mathbf{x})\right] \\ &\quad + \mathbb{E}_\mathbf{y}\left[\log(1 - D_\mathbf{X}(G_\mathbf{X}(\mathbf{y})))\right], \end{aligned} \tag{2}$$

in which $G_\mathbf{Y}$ learns to generate images $G_\mathbf{Y}(\mathbf{x})$ matching those from domain $\mathbf{Y}$, while $G_\mathbf{X}$ tries translating $\mathbf{y}$ to images from domain $\mathbf{X}$. Following *CycleGAN*, we also add the *cycle consistency loss* to constrain those mappings:

$$\mathcal{L}_{\text{cyc}_\mathbf{Y}}(G_\mathbf{X}, G_\mathbf{Y}; \mathbf{Y}) = \mathbb{E}_\mathbf{y}\left[||G_\mathbf{Y}(G_\mathbf{X}(\mathbf{y})) - \mathbf{y}||_1\right], \tag{3}$$

$$\mathcal{L}_{\text{cyc}_\mathbf{X}}(G_\mathbf{Y}, G_\mathbf{X}; \mathbf{X}) = \mathbb{E}_\mathbf{x}\left[||G_\mathbf{X}(G_\mathbf{Y}(\mathbf{x})) - \mathbf{x}||_1\right]. \tag{4}$$

3.3 Semantic Loss

Since our translation domains of interest are between synthetic images and real-world sensor images, we take advantage of the fact that many recent robotic simulators provide ground truth semantic labels and add the semantic constraint inspired by *CyCADA* [Hoffman *et al.*, 2017].

Assuming that for images from domain $\mathbf{X}$, the ground truth semantic information $S_{\mathbf{X}}$ is available, a semantic segmentation network $f_{\mathbf{X}}$ can be easily obtained by minimizing the *cross-entropy* loss, denoted $\text{CrossEnt}(S_{\mathbf{X}}, f_{\mathbf{X}}(\mathbf{X}))$.

We further assume that the ground truth semantic for domain $\mathbf{Y}$ is lacking (which is the case for most real scenarios), meaning that $f_{\mathbf{Y}}$ is not easily accessible. In this case, we provide "semi" semantic supervision to the training agents.

After $f_{\mathbf{X}}$ for semantic segmentaion of domain $\mathbf{X}$ is obtained, "semi" semantic supervision for the generators can be incorporated, by imposing consistency between the semantic map of the input and that of the generated output. This semantically consistent image translation can be achieved by minimizing the following losses (we use $f_{\mathbf{X}}$ to also generate "semi" semantic labels for domain $\mathbf{Y}$):

$$\mathcal{L}_{\text{sem}_{\mathbf{Y}}}(G_{\mathbf{Y}}; \mathbf{X}, f_{\mathbf{X}}) = \text{CrossEnt}(f_{\mathbf{X}}(\mathbf{X}), f_{\mathbf{X}}(G_{\mathbf{Y}}(\mathbf{X}))) \quad (5)$$

$$\mathcal{L}_{\text{sem}_{\mathbf{X}}}(G_{\mathbf{X}}; \mathbf{Y}, f_{\mathbf{X}}) = \text{CrossEnt}(f_{\mathbf{X}}(\mathbf{Y}), f_{\mathbf{X}}(G_{\mathbf{X}}(\mathbf{Y}))) \quad (6)$$

3.4 *Shift Loss* for Consistent Generation

Different from the current literature for *image-to-image translation* or *domain adaptation*, our model is additionally expected to output consistent images for sequential input data. Although by adding $\mathcal{L}_{\text{sem}}$, the semantics of the consecutive outputs are constrained, inconsistences and artifacts still occur quite often. Moreover, in cases where ground truth semantics are unavailable from either domain, the sequential outputs are even less constrained, which could potentially lead to inconsistent DRL policy outputs. To constrain the consistency even in these situations, following the discussion from Sec. 2.3, we introduce *shift loss* below.

For an input image $\mathbf{x}$, we use $\mathbf{x}_{[x\to i, y\to j]}$ to denote the result of a shift operation: shifting $\mathbf{x}$ along the X axis by i pixels, and j pixels along the Y axis. We sometimes omit $y \to 0$ or $x \to 0$ in the subscript if the image is only shifted along the X or Y axis.

According to [Ruder *et al.*, 2017], a trained stylization network is shift invariant to shifts of multiples of $\mathbf{s}$ pixels ($\mathbf{s}$ represents the total downsampling factor of the network), but can output significantly different stylizations otherwise. This causes the output of trained network to change greatly for even very small changes in the input. We thus propose to add a conceptually simple yet direct and effective *shift loss*:

$$\mathcal{L}_{\text{shift}_{\mathbf{Y}}}(G_{\mathbf{Y}}; \mathbf{X}) = \mathbb{E}_{\mathbf{x},\, i,j\sim u(1,\mathbf{s}-1)} \left[\left|\left| G_{\mathbf{Y}}(\mathbf{x})_{[x\to i, y\to j]} - G_{\mathbf{Y}}(\mathbf{x}_{[x\to i, y\to j]}) \right|\right|_2^2 \right], \quad (7)$$

$$\mathcal{L}_{\text{shift}_{\mathbf{X}}}(G_{\mathbf{X}}; \mathbf{Y}) = \mathbb{E}_{\mathbf{y},\, i,j\sim u(1,\mathbf{s}-1)} \left[\left|\left| G_{\mathbf{X}}(\mathbf{y})_{[x\to i, y\to j]} - G_{\mathbf{X}}(\mathbf{y}_{[x\to i, y\to j]}) \right|\right|_2^2 \right], \quad (8)$$

where u denotes the uniform distribution.

Shift loss constrains the shifted output to match the output of the shifted input, regarding the shifts as image-scale movements. under the assumption that only limited regional movement would appear in subsequent input frames, *shift loss* effectively smoothes the mapping function for small regional movements, restricting the changes in its outputs for subsequent inputs frames. It can be regarded as a cheap alternative for imposing consistency constraints on small movements, eliminating the need for the relatively expensive optical flow information, which is crucial for meeting the requirement of real-time control in robotics.

3.5 Full Objective

Our full objective for learning *VR Goggles* (Fig. 1) is:

$$\begin{aligned}\mathcal{L}(G_{\mathbf{Y}}, G_{\mathbf{X}}, D_{\mathbf{Y}}, D_{\mathbf{X}}; \mathbf{X}, \mathbf{Y}, f_{\mathbf{X}}) = & \quad (9)\\ & \mathcal{L}_{\text{GAN}_{\mathbf{Y}}}(G_{\mathbf{Y}}, D_{\mathbf{Y}}; \mathbf{X}, \mathbf{Y}) + \mathcal{L}_{\text{GAN}_{\mathbf{X}}}(G_{\mathbf{X}}, D_{\mathbf{X}}; \mathbf{Y}, \mathbf{X})\\ & + \lambda_{\text{cyc}} \left(\mathcal{L}_{\text{cyc}_{\mathbf{Y}}}(G_{\mathbf{X}}, G_{\mathbf{Y}}; \mathbf{Y}) + \mathcal{L}_{\text{cyc}_{\mathbf{X}}}(G_{\mathbf{Y}}, G_{\mathbf{X}}; \mathbf{X})\right)\\ & + \lambda_{\text{sem}} \left(\mathcal{L}_{\text{sem}_{\mathbf{Y}}}(G_{\mathbf{Y}}; \mathbf{X}, f_{\mathbf{X}}) + \mathcal{L}_{\text{sem}_{\mathbf{X}}}(G_{\mathbf{X}}; \mathbf{Y}, f_{\mathbf{X}})\right)\\ & + \lambda_{\text{shift}} \left(\mathcal{L}_{\text{shift}_{\mathbf{Y}}}(G_{\mathbf{Y}}; \mathbf{X}) + \mathcal{L}_{\text{shift}_{\mathbf{X}}}(G_{\mathbf{X}}; \mathbf{Y})\right),\end{aligned}$$

where λ_{cyc}, λ_{sem} and λ_{shift} controls the weighting for each loss. This corresponds to solving the following optimization:

$$G^*_{\mathbf{Y}}, G^*_{\mathbf{X}} = \arg \min_{G_{\mathbf{Y}}, G_{\mathbf{X}}} \max_{D_{\mathbf{Y}}, D_{\mathbf{X}}} \mathcal{L}(G_{\mathbf{Y}}, G_{\mathbf{X}}, D_{\mathbf{Y}}, D_{\mathbf{X}}). \quad (10)$$

4 Experiments

4.1 *Artistic Style Transfer* for Videos

To evaluate our method, we firstly conduct experiments for *artistic style transfer* for video sequences, to validate the effectiveness of *shift loss* on constraining consistency for sequential frames. We collect a training dataset of 98 HD video footage sequences (from *VIDEVO*[1], containing 2450 frames in total); the *Sintel*[2] sequences are used for testing, as their ground-truth optical flow is available. We compare the performance of the models trained under the following setups:

- *FF* [Johnson *et al.*, 2016]: Conanical feed forward style transfer trained on single frames;
- *FF+flow* [Huang *et al.*, 2017]: *FF* trained on sequential images, with optical flow added for imposing temporal constraints on subsequent frames.
- *Ours*: *FF* trained on single frames, with an additional *shift loss*, as discussed in Sec. 3.4.

We do not compare our method with that of [Ruder *et al.*, 2017], as they require optical flow as input during deployment. This is relatively expensive for our target application of real-time control.

Implementation-wise, we use the pretrained *VGG-19* as the loss network, *relu*2_2 as the content layer, *relu*1_2, *relu*2_2, *relu*3_2 and *relu*4_2 as the style layers. We set the weight for each loss as follows: 1e5 for content, 2 for style, 1e-7 for spatial regularization, 10 for optical flow, and 100 for shift. The downsampling factor $\mathbf{s}$ for our transformer network is 4; we use the same transformer network architecture and style images as in [Johnson *et al.*, 2016]. Shifts are uniformly sampled from $[1, \mathbf{s}-1]$ for every training frame.

As a proof of concept, we begin our evaluation by comparing the setups on their ability to generate shift invariant stylizations. In particular, for each image $\mathbf{x}$ in the testing dataset, we generate 4 more test images by shifting the original image along the X axis by $1, 2, 3, 4$ pixels respectively, and pass all 5 frames ($\mathbf{x}$, $\mathbf{x}_{[\mathbf{x}\to 1]}$, $\mathbf{x}_{[\mathbf{x}\to 2]}$, $\mathbf{x}_{[\mathbf{x}\to 3]}$, $\mathbf{x}_{[\mathbf{x}\to 4]}$) through the trained network to examine the consistency of the generated images (Fig. 2).

[1]https://www.videvo.net/ [2]http://sintel.is.tue.mpg.de/

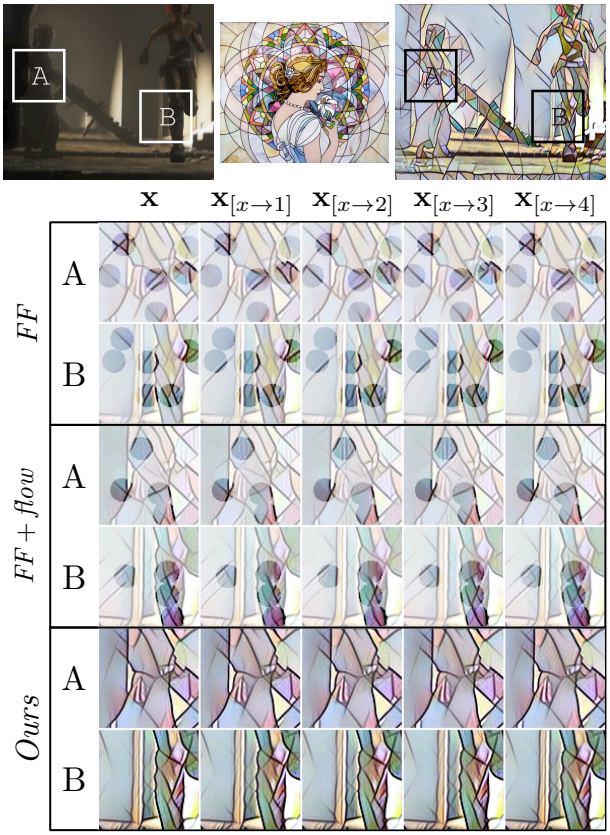

Figure 2: Shift-invariance evaluation, comparing between *FF*, *FF+flow* and *Ours*. We shift an input image $\mathbf{x}$ along the X axis by $1, 2, 3, 4$ pixels respectively and feed all 5 frames through the networks trained via *FF*, *FF+flow* and *Ours*, and show the generated stylizations. We mark the most visible differences in small circles and dim the rest of the generated images. As is discussed in [Ruder *et al.*, 2017], *FF* generates almost identical stylizations for $\mathbf{x}$ and $\mathbf{x}_{[x\to4]}$ (because 4 is a multiple of the total downsampling factor of the trained network), but those for $\mathbf{x}_{[x\to1]}$,$\mathbf{x}_{[x\to2]}$,$\mathbf{x}_{[x\to3]}$ differ significantly. *FF+flow* improves the shift-invariance, but we suspect the improvement is due to the inexplicit consistency constraint on regional shifts imposed by optical flow. *Ours*, is able to generate shift-invariant stylizations with the proposed *shift loss*.

The results shown in Fig. 2 validate the discussion from [Ruder *et al.*, 2017], since the stylizations for $\mathbf{x}$ and $\mathbf{x}_{[x\to4]}$ from *FF* are almost identical ($\mathbf{s} = 4$ for the trained network), but differ significantly otherwise. *FF-flow* improves the invariance by a limited amount; *Ours* method is capable of generating consistent stylizations for shifted input frames, with the *shift loss* directly reducing the shift variance.

We continue by evaluating the consistency of the stylized sequential input frames. In Fig. 3, we show the *temporal error maps*, the same metric as in [Huang *et al.*, 2017], of two stylized consecutive frames for each method. *Ours* (bottom row) achieves the highest temporal consistency.

Furthermore, we evaluate the temporal loss computed using the ground truth optical flow for the *Sintel* sequences (Table 1). Although the temporal loss is part of the optimization objective of *FF-flow*, and our method does not have access to any optical flow information, *Ours* is still able to achieve lower temporal loss with the *shift loss* constraint.

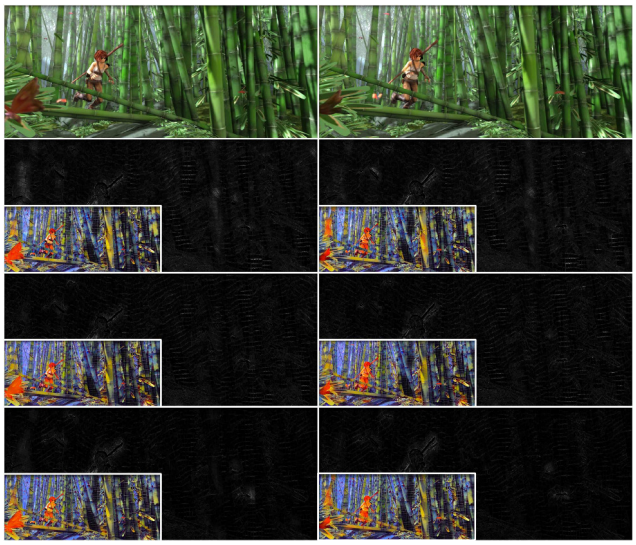

Figure 3: *Temporal error maps* between generated stylizations for subsequence input frames. $1st$ *row*: input frames; $2nd \sim 4th$ *row*: temporal error maps (with the corresponding stylizations shown on top) of outputs from *FF*, *FF+flow*, and *Ours*. We here choose a very challenging style (*mosaic*) for temporal consistency, as it contains many fine details, with tiny tiles laid over the original image in the final stylizations. Yet, *Ours* achieves very high consistency.

4.2 *Domain Adaptation* for *Outdoor* Scenarios

Next we validate the *shift loss* in the field of *domain adaptation*, firstly in *outdoor* urban street scenarios (where we collect synthetic domain images $\mathbf{X} \sim p_{\text{sim}}$ from the *CARLA* simulator [Dosovitskiy *et al.*, 2017], and realistic domain images $\mathbf{Y} \sim p_{\text{real}}$ from the *RobotCar* dataset [Maddern *et al.*, 2017]). We compare the following three setups:

- *CyCADA* [Hoffman *et al.*, 2017]: *CycleGAN* with semantic constraints, trained on single frames;
- *CyCADA+flow*: *CyCADA* with the temporal constraint as in [Huang *et al.*, 2017], trained on sequential frames;
- *Ours*: *CyCADA* with *shift loss*, trained on single frames; we refer to this as *VR Goggles*.

Table 1: Temporal loss comparison between *FF*, *FF+flow* and *Ours*. This metric is part of the optimization objective of *FF+flow*, while optical flow is never provided to *Ours*; yet our method is able to achieve lower temporal loss on the evaluated *Sintel* sequences.

	mosaic			*lamuse*		
	FF	*FF+flow*	*Ours*	*FF*	*FF+flow*	*Ours*
ambush5	0.152	0.130	**0.127**	0.154	0.131	**0.130**
bamboo1	0.119	0.093	**0.086**	0.123	0.097	**0.090**
market6	0.132	0.110	**0.108**	0.135	**0.112**	**0.112**
temple2	0.127	0.104	**0.098**	0.138	0.108	**0.107**
sleeping2	0.115	0.089	**0.083**	0.121	0.100	**0.092**
shaman3	0.124	0.095	**0.087**	0.132	0.106	**0.094**
alley2	0.122	0.096	**0.090**	0.129	0.104	**0.094**
bamboo2	0.113	0.091	**0.089**	0.114	**0.090**	0.091
alley1	0.113	0.085	**0.078**	0.127	0.096	**0.083**
sleeping1	0.125	0.099	**0.092**	0.132	0.102	**0.101**

Figure 4: Comparison of the translated images for sequential input frames for the different approaches. $1st$ *row*: two subsequent input frames from the realistic domain, with several representative images from the simulated domain shown in between; $2nd \sim 4th$ *row*: outputs from *CyCADA*, *CyCADA+flow* and *Ours*. Our method is able to output consistent subsequent frames and eliminate artifacts. We adjust the brightness of some zoom-ins for visualization purposes.

We pretrain the segmentation network $f_{\mathbf{X}}$ using *Deeplab* [Chen *et al.*, 2017]. It is worth mentioning that the original *CyCADA* paper did not use the semantic constraint in their experiments due to memory issues. We are able to incorporate semantic loss calculation, by cropping the input images. Actually, a naive random crop would highly likely lead to semantic permutations; so we crop inputs of the two domains in the same training iteration from the same random position, and our empirical results show that this greatly stabilizes the adaptation. The input images are of size 450×800, we train the network with 256×256 crops. We use the same network architecture as in *CycleGAN*, and train for 50 epochs with a learning rate of $2e-4$, as we observe no performance gain training for longer iterations.

In Fig. 4, we show a comparison of the subsequent frames generated by the three approaches. Our method again achieves the highest consistency and eliminates more artifacts due to the smoothness of the learned model.

4.3 *Domain Adaptation* for *Indoor* Scenarios with Real-world Robotic Experiments

Finally, we conduct *domain adaptation* for *indoor* office environments ($\mathbf{X} \sim p_{\text{sim}}$ rendered from a self-built *Gazebo* [Koenig *et al.*, 2004] world and $\mathbf{Y} \sim p_{\text{real}}$ captured from a real office, using a *RealSense R200 camera* mounted on a *Turtlebot3 Waffle*). We validate our proposed method of using the *VR Goggles* to facilitate the transfer of polices trained in the simulated domain to the realistic domain, with a set of real-world robotic experiments.

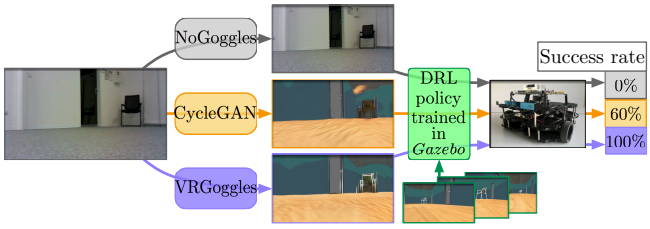

Figure 5: Real-world visual control experiment. A DRL agent is trained in a simulated office environment, that is able to navigate to chairs based on visual input. Without retraining or finetuning the DRL policy, our proposed *VR Goggles* enables the mobile robot to directly deploy this policy in real office environments, achieving 100% success rate in a set of real-world experiments. We refer to the attached video for details of the real-world experiments.

Specifically, we begin by training a DRL agent in a simulated office environment (A3C [Mnih *et al.*, 2016], $20k$ rollouts, 8 parallel CPU threads), to accomplish the task of navigating to chairs based purely on its front-facing color camera readings; the agent obtained a reward of -0.005 for a step cost, -0.05 for collision, and 1 for reaching the target. Then, we deploy the trained DRL policy onto a real robot in a real-world office, and compare the following adaptations:

- *NoGoggles*: Feed the sensor readings directly to the trained DRL policy;
- *CycleGAN* [Zhu *et al.*, 2017a]: Use *CycleGAN* to translate the real sensory inputs to the synthetic domain before feeding to the DRL policy; since the synthetic domain here (*Gazebo*) does not provide ground truth semantics, we drop the semantic constraint $\mathcal{L}_{\text{sem}}$;
- *Ours*: Use models trained by *CycleGAN* + *shift loss* as the *VR Goggles* to translate the input images.

We use the same network configuration as in Sec. 4.2, except that here the input images are of size 360×640.

We show in the attached video that, without *domain adaptation*, directly deploying the DRL policy fails completely in the real-world tasks; our proposed method achieves the highest success rate (0%, 60% and 100% for *NoGoggles*, *CycleGAN* and *Ours* respectively) due to the quality and consistency of the translated streams. The control cycle runs in real-time at 13Hz on a *Nvidia Jetson TX2*. In the video, we also show that VR Goggles can easily train a new model for a new type of chair without any adjustment to the previously trained control policy. We limited the velocity of the robot due to the camera exposure time, since motion blur can greatly influence the adaptation quality. We leave it as future work to evaluate on more compatible platforms.

5 Conclusions

To conclude, we tackle the *reality gap* when deploying DRL visual control policies trained in simulation to the real world, by translating the real image streams back to the synthetic domain during the deployment phase. Due to the sequential nature of the incoming sensor streams for control tasks, we propose *shift loss* to increase the consistency of the translated subsequent frames. We validate the *shift loss* in both *artistic*

style transfer for videos, and *domain adaptation*. In the end, we successfully verify our *domain adaptation* method for visual control through a set of real-world robot experiments.

Several future works can be conducted based on our method. For example, training DRL agents on more complicated tasks, and in more complicated simulated environments that provide ground truth semantic labels, such as the newly released *MINOS* [Savva *et al.*, 2017]. Also, since in this paper we have mainly focused on learning based visual navigation, applying our method to manipulation tasks would be an interesting direction.

References

[Bousmalis *et al.*, 2017] Konstantinos Bousmalis, Alex Irpan, Paul Wohlhart, Yunfei Bai, Matthew Kelcey, Mrinal Kalakrishnan, Laura Downs, Julian Ibarz, Peter Pastor, Kurt Konolige, et al. Using simulation and domain adaptation to improve efficiency of deep robotic grasping. *arXiv preprint arXiv:1709.07857*, 2017.

[Chen *et al.*, 2017] LC Chen, G Papandreou, I Kokkinos, K Murphy, and AL Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 2017.

[Dosovitskiy *et al.*, 2017] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on Robot Learning*, pages 1–16, 2017.

[Hoffman *et al.*, 2017] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. *arXiv preprint arXiv:1711.03213*, 2017.

[Huang *et al.*, 2017] Haozhi Huang, Hao Wang, Wenhan Luo, Lin Ma, Wenhao Jiang, Xiaolong Zhu, Zhifeng Li, and Wei Liu. Real-time neural style transfer for videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 783–791, 2017.

[Johnson *et al.*, 2016] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711. Springer, 2016.

[Koenig *et al.*, 2004] Nathan Koenig, B A, and Andrew Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 3, pages 2149–2154. IEEE, 2004.

[Lillicrap *et al.*, 2015] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[Maddern *et al.*, 2017] Will Maddern, Geoff Pascoe, Chris Linegar, and Paul Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research (IJRR)*, 36(1):3–15, 2017.

[Mnih *et al.*, 2015] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

[Mnih *et al.*, 2016] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, pages 1928–1937, 2016.

[Ruder *et al.*, 2017] Manuel Ruder, Alexey Dosovitskiy, and Thomas Brox. Artistic style transfer for videos and spherical images. *arXiv preprint arXiv:1708.04538*, 2017.

[Rusu *et al.*, 2017] Andrei A Rusu, Matej Večerík, Thomas Rothörl, Nicolas Heess, Razvan Pascanu, and Raia Hadsell. Sim-to-real robot learning from pixels with progressive nets. In *Conference on Robot Learning*, pages 262–270, 2017.

[Savva *et al.*, 2017] Manolis Savva, Angel X Chang, Alexey Dosovitskiy, Thomas Funkhouser, and Vladlen Koltun. Minos: Multimodal indoor simulator for navigation in complex environments. *arXiv preprint arXiv:1712.03931*, 2017.

[Schulman *et al.*, 2015] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, pages 1889–1897, 2015.

[Schulman *et al.*, 2017] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[Tai *et al.*, 2017] Lei Tai, Giuseppe Paolo, and Ming Liu. Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 31–36, Sept 2017.

[Tai *et al.*, 2018] Lei Tai, Jingwei Zhang, Ming Liu, and Wolfram Burgard. Socially-compliant navigation through raw depth inputs with generative adversarial imitation learning. In *Robotics and Automation (ICRA), 2018 IEEE International Conference on*, May 2018.

[Tobin *et al.*, 2017] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, pages 23–30. IEEE, 2017.

[Zhang *et al.*, 2017a] Jingwei Zhang, Jost Tobias Springenberg, Joschka Boedecker, and Wolfram Burgard. Deep reinforcement learning with successor features for navigation across similar environments. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2371–2378, Sept 2017.

[Zhang *et al.*, 2017b] Jingwei Zhang, Lei Tai, Joschka Boedecker, Wolfram Burgard, and Ming Liu. Neural slam. arxiv preprint. *arXiv preprint arXiv:1706.09520*, 3, 2017.

[Zhu *et al.*, 2017a] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2223–2232, 2017.

[Zhu *et al.*, 2017b] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali

Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 3357–3364. IEEE, 2017.