2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Practical Visual Deep Imitation Learning via Task-Level Domain Consistency

Mohi Khansari[1,*], Daniel Ho[2,*], Yuqing Du[2,3,*], Armando Fuentes[1], Matthew Bennice[1], Nicolas Sievers[2], Sean Kirmani[1], Yunfei Bai[2], Eric Jang[4]

***Abstract*— Recent work in visual end-to-end learning for robotics has shown the promise of imitation learning across a variety of tasks. Such approaches are however expensive both because they require large amounts of real world data and rely on time-consuming real-world evaluations to identify the best model for deployment. These challenges can be mitigated by using simulation evaluations to identify high performing policies. However, this introduces the well-known "reality gap" problem, where simulator inaccuracies decorrelate performance in simulation from that of reality. In this paper, we build on top of prior work in GAN-based domain adaptation and introduce the notion of a *Task Consistency Loss (TCL)*, a self-supervised loss that encourages sim and real alignment both at the feature and action-prediction levels. We demonstrate the effectiveness of our approach by teaching a 9-DoF mobile manipulator to perform the challenging task of latched door opening purely from visual inputs such as RGB and depth images. We achieve 69% success across twenty seen and unseen meeting rooms using only $\sim$16.2 hours of teleoperated demonstrations in sim and real. To the best of our knowledge, this is the first work to tackle latched door opening from a purely end-to-end learning approach, where the task of navigation and manipulation are jointly modeled by a single neural network.**

I. Introduction

In recent years, the field of vision-based robotics has seen significant developments in navigation [1], [2], [3] or manipulation [4], [5] separately. However, if we eventually seek to deploy robots in human environments, we require agents capable of doing both simultaneously [6], [7]. Most prior work in vision-based manipulation focuses on fixed scenes from a third person perspective, but mobile manipulation introduces the challenge of precisely coordinating base and arm motions. Furthermore, manipulating objects from egocentric vision necessitates generalization to much greater visual diversity, since the robot's view is continuously changing as it moves through the environment.

We choose to tackle this problem with imitation learning (IL), as recent work on end-to-end learning for manipulation has shown promising results with this approach [8], [9], [10]. However, imitation learning from raw sensor outputs requires numerous real world demonstrations. These demonstrations can be expensive and time consuming to collect, especially with the more complex action space of a mobile manipulator. Even after acquiring this data, evaluating learned policies in reality for generalization across a wide variety of unseen situations can still be time-consuming and hazardous. Unlike perception benchmarks, where validation datasets inform model selection, error on offline expert trajectories in robotics does not necessarily inform how the policy will behave if it drifts away from expert trajectories.

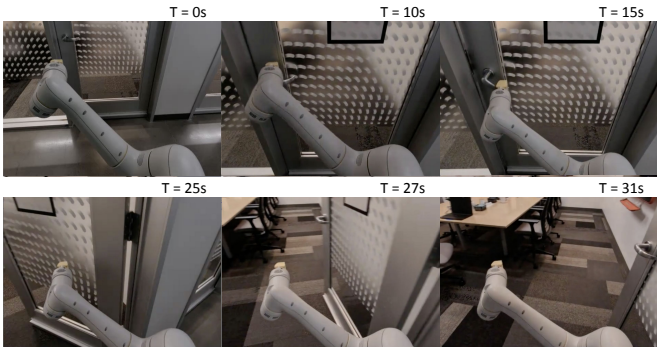

Fig. 1. A sample door opening trajectory in a real world office environment using our method. The robot navigates to the door from 0-10s, unlatches the door from 10-20s, then fully opens the door and enters the room from 20-31s.

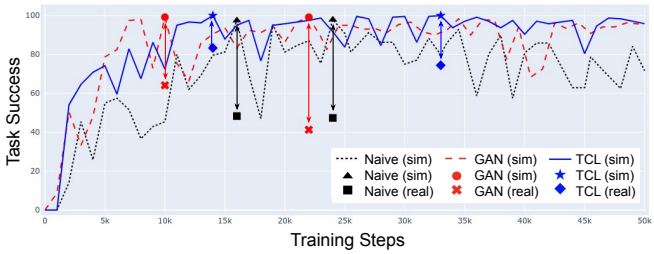

Fig. 2. Matching sim and real evaluation is crucial for real-world model deployment in a cost-effective manner. Our method, TCL, outperforms the baselines (mixing sim + real data (Naive) and sim + real + GAN-adapted sim data (GAN)) by reducing the sim-real gap from $+45\%$ to 21.1%.

Simulators are often used to alleviate challenges with data collection and evaluation. The sim-to-real community often focuses on the ability to generate plentiful training data in simulation, but we posit that gathering enough real data to learn good policies is not too difficult; *what is often far more time-consuming are the number of real-world trials needed to accurately compare policies across a number of generalization settings*. Policies trained and evaluated in simulation suffer from the well known "reality gap", where visual and physical inaccuracies in the simulator can cause a high performing policy in simulation to still under-perform in the real world (see Figure 2). In order to scale robotics to many real-world scenarios, we require reliable simulated evaluations that are representative of real-world performance.

One popular and simple approach to bridging the reality gap is "domain randomization" [11], [12], where a known set of simulator parameters, such as object textures and

[1]Everyday Robots, [2]Work done while at Everyday Robots,
[3]UC Berkeley, [4]Work done while at Google, *Major contributors

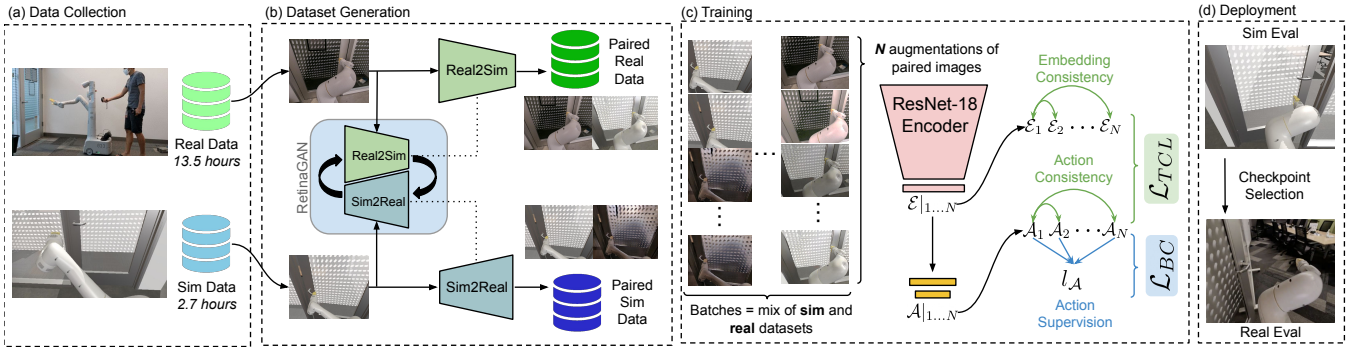

Fig. 3. **(a, b)** We collect a set of sim and real images through teleoperation, and use them first to train a RetinaGAN model. **(b)** We then use the trained real-to-sim and sim-to-real models to create paired (real, adapted real) and paired (sim, adapted sim) images, respectively. **(c)** Policy representations and predictions are encouraged to be invariant between paired images via a novel Task Consistency Loss. The same procedure can be used for the depth images (not shown here but used in the paper). **(d)** We use simulation in parallel to model training to evaluate all the checkpoints to help with the best checkpoint selection for the real world deployment.

joint stiffness coefficients, are randomized within a pre-specified range. Another approach is "domain adaptation", where the goal is to learn features and predictions invariant to the domain of model inputs. We build on past work in CycleGAN-based domain adaptation [13] by introducing additional feature-level and prediction-level alignment losses, the *Task Consistency Loss*, between the adapted sim-to-real and real-to-sim images. We also extend our domain adaptation approach to the depth modality, showing our method can work with RGB, depth, and RGB-D inputs. Thus we leverage observations collected in both sim and reality for not just IL, but also for domain adaptation.

To test our approach, we focus on a challenging mobile manipulation task: *latched door opening*. A mobile manipulator robot with head-mounted RGB-D sensors must autonomously approach a door, use the arm to turn the door handle, push the door open, and enter the room (Figure 1). Prior work on door opening decouples the manipulation behavior from the navigation behavior, by first localizing the handle, planning an approach, then executing a grasping primitive [14]. In contrast, our method solely uses egocentric RGB-D images from the camera on the robot head and a single neural network for coordinating both arm and base motion to successfully open a variety of doors in an office building. In this paper, we will present an imitation learning system for mobile manipulation with a novel domain adaptation approach for aligning simulated and real performance. Our key contributions are:

1) Introducing feature-level and action-level sim and real alignment from a novel Task Consistency Loss, in addition to image-level alignment from modality-specific GANs. As shown in Figure 2, our method outperforms existing baselines of naively mixing real and sim and prior methods of GAN-adapted sim by a substantial margin of $+12$ percentage-point.
2) Deployment of a difficult mobile manipulation task in natural realistic environment across two buildings at Alphabet, and achieving 69% success on twenty meeting rooms (6 seen and 14 unseen during the training), with only 13.5 hours of real demonstrations and 2.7 hours of simulated demonstrations.

II. Related Work

Deep Learning for Mobile Manipulation: Although significant progress has been made in robot navigation and manipulation tasks individually, tackling the intersection of the two with deep learning is still relatively under-explored. Recent work has developed reinforcement learning methods for mobile manipulators, but are either only evaluated in simulation [6] or require many hours of real world learning [15], [16]. The work by [17] proposes a hierarchical reinforcement learning approach for mobile manipulation tasks, but tackles a simpler variant of door opening, where the door opens by pushing a button or the door directly. [10] uses end-to-end imitation learning to push open swing doors (no handle) by driving the base of a mobile manipulator with the arm fixed. They improve performance in real by concatenating sim demonstrations and sim-to-real adapted images to the real demonstration dataset, but do not directly tackle the the gap between simulation and real world performance. We introduce a Task Consistency Loss to address that limitation, which enables us to scale end-to-end imitation learning to the harder task of latched door opening.

A range of robotic control approaches have been proposed specifically for door opening, but require identifying the door handle through human intervention [18] or additional sensor instrumentation [19], [20], [21], [22], [23]. For instance, [14] uses an object detector to identify the door handle and a scripted controller to grasp the handle to open the door. In contrast, our approach is fully end-to-end: navigation and manipulation decisions are inferred from first-person camera images without hand-engineering of object or task representations.

Sim-to-real Transfer: Prior work in sim-to-real transfer falls broadly in three categories: domain adaptation, domain randomization, and system identification. Our work focuses on domain adaptation, whereby discrepancies between sim and real are directly minimized. This could happen on the *pixel*-level, where synthetic images are stylistically translated to appear more realistic, or on a *feature*-level, where deep neural network features from simulation and real inputs are optimized to be similar. Pixel-level domain adaptation

work commonly make use of generative models to transfer inputs between domains, especially Generative Adversarial Networks (GANs) [24]. In robotics, this is frequently applied to robotic manipulation and grasping [25], [26]. Among these, RetinaGAN [13] translates images using perception-consistency to preserve object semantics and structure. RL-CycleGAN [27] trains CycleGAN [28] jointly with a reinforcement learning (RL) model. Here, consistency of RL predictions before and after GAN adaptation preserves visual qualities deemed important to RL learning. Our work leverage these methods to apply a notion of consistency to further reduce the sim-real gap for the purpose checkpoint selection for real-world model deployment.

Feature-level domain adaptation work commonly analyze the *distribution* of features from sim and real domains at the batch-level. DANN and DSN [29], [30] adversarially teach a network to extract features which does not discriminate between sim and real domains. Our feature-level domain adaptation method falls under self-supervised representation learning, which is commonly faciliated by increasing similarity between embeddings of positive image pairs. Prior work in this area has proposed using pairs generated from augmentations (e.g. random crop, flip, patch, colour shift) [31], [32], [33]. We extend this approach to aligning paired simulated and real images from pixel-level domain adaptation GANs. That is, we maximize similarity between embeddings of the pairs (original sim, adapted sim) and (original real, adapted real).

Sim-to-real methods are utilized in mediated perception tasks in robotics, such as segmentation for autonomous driving [34] or pose estimation for object manipulation [35]. Because these tasks decouple perception from control, performance on real data are cheaply evaluated via metrics like IoU and AUROC on offline real data. However, evaluating end-to-end robot policies cannot be trivially done offline, and thus requires running multi-step predictions in the real world due to the causality effects (the current action can affect future observations, and future observations can further affect the proceeding actions). While our method can help with leveraging the simulation data for policy training similarly to previous domain adaptation works, it is additionally designed to help mitigate the cost of expensive real-world evaluation for end-to-end policies. One desideratum of our method is that simulated evaluation performance corresponds tightly to real world performance, and that this is achieved without much real-world tuning.

Multimodal Learning: Prior work in manipulation policies often use the RGB image alone as input. More recently, there's been a movement to use other modalities—such as depth, optical flow, and semantic segmentation [36], [37], [38], [39], [40]—to improve sample efficiency and final performance of manipulation policies. While these derived higher-level modalities can implicitly be learned from the RGB image alone, using these geometric, semantic, and motion cues can improve training speed and task performance without the burden of learning from scratch.

III. Problem Setup

A. Imitation Learning

Our goal is to learn a policy, $\pi(a|s)$, that outputs a continuous action $a \in \mathcal{A}$ given an image $s \in \mathcal{S}$ which may be RGB, depth, or both. In imitation learning, we assume we have a dataset of expert demonstrations $\tau^* = (s_0, a_0, s_1, a_1, ..., s_{T-1}, a_{t-1}, s_T)$ with the actions generated by an expert policy π^*. We then learn to imitate this dataset with behaviour cloning, where the objective is to minimize a divergence between $\pi(a|s)$ and $\pi^*(a|s)$ given the same state s. Common minimization objectives are negative log-likelihood or mean-squared error.

B. Task

We consider the task of latched door opening in a real office environment, in which the robot needs to drive a distance of $\sim 1m$ to bring the arm in close vicinity of the door handle, use the arm to rotate the handle, and then use coordinated base and arm motions to swing the door open. This task has the following challenges:

1) **High dimensional action space:** The task is only feasible by moving both the robot base (2-DoF) and the arm (7-DoF). A 9-dimensional action space together with high-dimensional visual inputs make this task particularly challenging for imitation learning, especially with a limited number of expert demonstrations.
2) **Mobile manipulation coordination:** The task requires precise coordination and time-synchronization between base and arm movements. For instance, there is no use in moving the arm if the handle is outside the robot's reachable space, and driving the base forward into a latched door leads to collision and robot arm breakage.
3) **Long horizon**: The task takes an expert 17 to 60 seconds to demonstrate, corresponding to up to 600 (input, action) pairs per episode. This long duration heightens task difficulty due to compounding errors associated with behavior cloning models [41].
4) **Bi-modal task nature**: We are training a single model to open both left-swing and right-swing doors, so the policy needs to infer the door swing direction and handle location from the image.

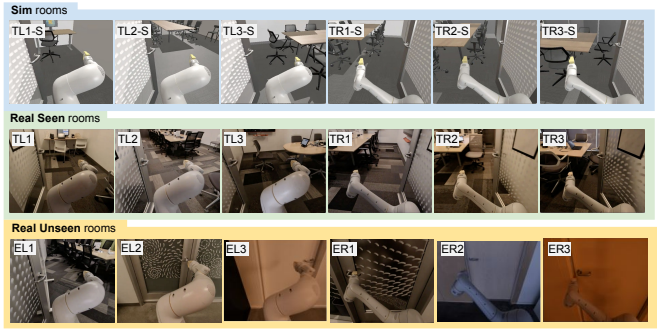

Fig. 4. Respectively from top to bottom: 6 snapshots of sim meeting room scenes, used for data collection and checkpoint evaluation; 6 real meeting rooms used for data collection; illustrating 6 out of 14 unseen meeting rooms used for evaluation.

C. Data Collection

We collect expert actions via teleoperation at 10Hz and record the corresponding RGB and depth image inputs. During the demonstration, the user can control both the robot base and arm via a teleoperation device.

1) Real Dataset: In total, we collected 2068 real world demonstrations (corresponding to $\sim$13.5 hours) across 6 meeting rooms (3 left-swing and 3 right-swing doors). For each episode, we position the robot in front of the meeting room $\sim$1 meter away from the door. We then randomize the initial pose $\delta x \sim U(-0.25, +0.25)$ meters, $\delta y \sim U(-0.1, +0.1)$ meters, and $\delta\psi \sim U(-5, +5)$ degrees, where x and y correspond to the axes orthogonal and parallel to the door respectively, ψ is the base orientation, and U is the uniform distribution function. After pose randomization, we move the arm to a predefined initial joint configuration using the robot's built-in controller. We use a different initial configuration for the left and right swing doors to make the task more kinematically tractable. This prior knowledge of swing direction used in setup is not passed to the model; hence the model has to infer this from images.

After initial setup, the expert commands the robot via a hand-held teleoperation device and completes the episode when the door is sufficiently open such that the robot can enter the room without collision. We do not control the condition of the room (light, chair, table, ...) and collect demonstrations in the natural state left by previous users.

2) Sim Dataset: We create 3D models of the 6 training meeting rooms with lower-fidelity textures but sufficient structural detail for the RetinaGAN domain adaptation model to translate to real (see Figure 4). During sim data collection, we use the same teleoperation interface, task setup, and success metric as in real. In total, we collected $\sim$500 demonstrations, corresponding to $\sim$2.7 hours of data.

IV. Method

Our method leverages the domain adaptation GAN works [13], [28] and extends them by further reducing the sim-to-real gap not only at the visual level, but also at the feature and action prediction level using the Task Consistency Loss (TCL). We use the following notation:

- Subscripts $_{RGB}$ and $_{D}$ reference parameters or functions associated with RGB and depth images, respectively.
- $\mathcal{I}$ refers to an input image, either RGB, $\mathcal{I}_{RGB} \in \mathcal{R}_{+}^{H\times W\times 3}$, or Depth, $\mathcal{I}_{D} \in \mathcal{R}_{+}^{H\times W}$.
- $\mathcal{D}$ references an image augmentation/distortion function. For RGB, $\mathcal{D}_{RGB}$, we apply random crop, brightness, saturation, hue, contrast, cutout, and additive Gaussian noise. For depth, $\mathcal{D}_{D}$, we only apply random crop and cutout.
- $\mathcal{G}$ refers to sim2real $\mathcal{G}^{sim2real}$ or real2sim $\mathcal{G}^{real2sim}$ generators of RetinaGAN or CycleGAN models. We use separate GANs for each modality. For example, $\mathcal{G}_{RGB}^{sim2real}$ transfers RGB images from the sim domain to the real domain.

For brevity, we may drop subscripts and superscripts to indicate that a process can be applied on either input modality. For instance, $\mathcal{I}$ indicates use of either RGB or depth images. Examples of transformed RGB and depth images through $\mathcal{D}$ and $\mathcal{G}^{sim2real}$ are shown in Figure 5.

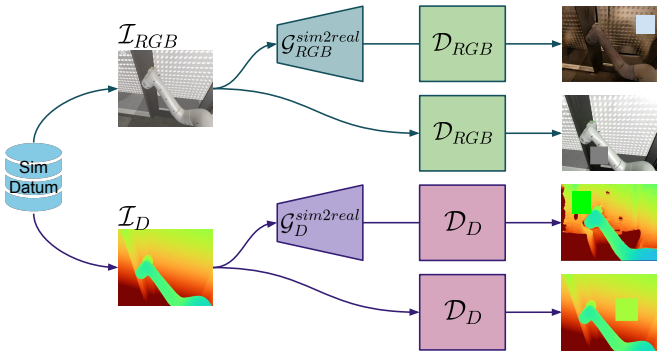

Fig. 5. Illustration of applying augmentations through $\mathcal{D}$ and $\mathcal{G}^{sim2real}$ to an input dataset from simulation. Turbo colormap applied to depth images for clarity. This process is reversed in GAN-translated real images (not shown in here).

A. Paired Image Generation using GANs

We visually align images from unpaired sim and real datasets by building on top of the pixel-level domain adaptation techniques, RetinaGAN [13] and CycleGAN [28], by extending them to the latched door opening task. From these models, we use the sim2real and real2sim generator networks to adapt images from our original demonstrations. The resulting datasets contain an original sim or real image and the corresponding domain-translated *paired* images.

RGB GAN: We train a GAN using the perception consistency loss based on Section V.C of the RetinaGAN work [13], re-using the off-the-shelf RetinaNet object detector trained on object grasping examples [42]. RetinaGAN trains unsupervised, using only images collected from teleoperation, described in Section III. Within GAN-translated RGB images of simulation, glass door patterns appear more translucent, lighting conditions more randomized, lighting effects like global illumination and ambient occlusion added, and color tones adjusted. This process is reversed in GAN-translated real images.

Depth GAN: For the depth modality, we train a CycleGAN [28] model—we lack a depth detector needed for RetinaGAN—on stereo real depth (computed using HitNet [43] stereo matching) and simulated ground truth depth images. We pre-process images by clipping depth to 10 meters. The trained model reliably translates between differences in the two domains. Foremost, real images have significant noise from sensors and stereo matching, while simulation images are noiseless. The glass and privacy film of the doors appear as opaque in simulation but translucent in real, where depth bleeds through to the floor of the conference room behind. The depth GAN learns to inpaint real image pixels which have passed through the door, and it generates patches of depth behind the glass in simulation images. Figure 5 shows an example of adapted sim images.

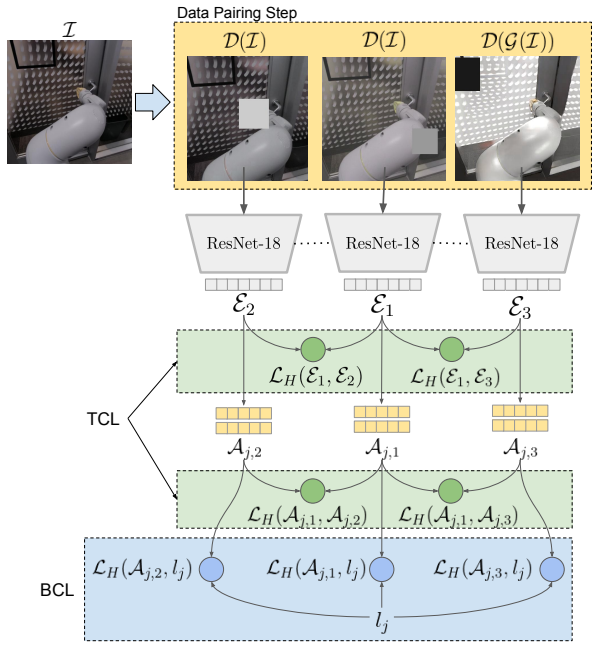

Fig. 6. Task Consistency Loss: We create pairs by 1) augmenting the image, and 2) adapting the image from sim-to-real or real-to-sim with the corresponding GAN, then applying augmentations. We pass all images of the same modality through the same ResNet-18 [44] encoder f_ϕ followed by a normalization layer to generate embeddings $\mathcal{E}_i$, and then pass them through a two layer MLP g_ϕ to get the predicted actions $\mathcal{A}_{j,i}$. Thus, for each image we can compute $\mathcal{L}_{TCL}$ and $\mathcal{L}_{BC}$, using $\mathcal{E}_i$, $\mathcal{A}_{j,i}$, $\forall i \in 1..N$ and $j \in (a, b, f)$, where $\mathcal{A}_b$, and $\mathcal{A}_f$ correspond to predicted actions for arm, base, and termination, respectively.

B. Task Consistency Loss (TCL)

In addition to adaptation at the pixel level through GANs, we introduce a novel auxiliary loss, TCL, to encourage stronger alignment between the sim and real domains for adaptation at the feature and the action-prediction levels. For a given image $\mathcal{I}$, we can generate N variations, $\mathcal{I}|_{1..N}$, by applying augmentations such as $\mathcal{D}$, $\mathcal{G}$, or both.

In this paper we consider the following three variations for an input image $\mathcal{I}$: (1) Original sim/real image distorted with $\mathcal{D}$, $\mathcal{I}_1 = \mathcal{D}(\mathcal{I})$, (2) A distorted instance of the original sim/real image, $\mathcal{I}_2 = \mathcal{D}(\mathcal{I})$. The consistency loss between $\mathcal{I}_1$ and $\mathcal{I}_2$ enforces invariancy with respect to image distortion transformations, and (3) Adapted original images via $\mathcal{G}$ followed by a distortion, $\mathcal{I}_3 = \mathcal{D}(\mathcal{G}(\mathcal{I}))$. The consistency loss between $\mathcal{I}_1$ and $\mathcal{I}_3$ enforces invariancy with respect to the domain transformation as well as the image distortions.

The N variations of the input image $\mathcal{I}|_{1..N}$ depict the same instant of time. Hence, the image embeddings $\mathcal{E}|_{1..N}$ and predicted actions $\mathcal{A}|_{1..N}$ should be invariant under augmentations $\mathcal{D}$ and $\mathcal{G}$, and we derive our self-supervised signal by enforcing this invariancy. We hypothesize that this will help close the sim-to-real gap and make performance in simulation more representative of that in reality. Additionally, imposing this consistency loss on images augmented with random cutout may improve robustness to occlusions; it encourages the model to learn features in context of other salient features (e.g. the handle based on the door frame, see Figure 6).

To calculate TCL, we pass all variations of the input image through the same network to calculate corresponding image embeddings $\mathcal{E}|_{1..N}$ and estimated actions $\mathcal{A}|_{1..N}$. Then, we apply a Huber loss $\mathcal{L}_H$ [45] to penalize discrepancies between pairs as follows:

$$\mathcal{L}_{TCL} = \sum_{i=2}^{N} \Big(\mathcal{L}_H(\mathcal{E}_1, \mathcal{E}_i) + \sum_{j \in (a,b,f)} \mathcal{L}_H(\mathcal{A}_{j,1}, \mathcal{A}_{j,i}) \Big) \quad (1)$$

where the first term imposes consistency loss over the embeddings and the second term penalizes estimated action errors between all variations. Note that $\mathcal{A}_a$, $\mathcal{A}_b$, and $\mathcal{A}_f$ correspond to predicted actions for arm, base, and termination, respectively. The augmentation and loss setup for the feature-level TCL is shown in Figure 6.

C. Behavior Cloning Loss (BCL)

The behavior cloning loss is applied at each network head to enforce similarity between predicted actions $\mathcal{A}_j$ and demonstrated labels l_j, $\forall j \in (a, b, f)$. We use the same label to calculate BCL for all N variations of the input image, which can further reinforce invariancy across applied image augmentations:$\mathcal{L}_{BC} = \sum_{j \in (a,b,f)} \sum_{i=1}^{N} \mathcal{L}_H(\mathcal{A}_{j,i}, l_j)$. The overall policy training loss used is: $\mathcal{L} = \mathcal{L}_{BC} + \mathcal{L}_{TCL}$

D. Multi-Sensor Network Architecture

We use the methods described in Section IV-A to generate domain adapted and augmented images for each modality, then apply TCL as described in Section IV-B. To combine the different modalities, we concatenate all permutations of the N different variations per modality to get N^2 RGB-D embeddings. Empirically, we find that sensor fusion at the embedding level leads to higher task success than channel-wise fusion of the raw RGB and depth images prior to passing to the ResNet-18 [44] encoders. We then pass the concatenated embeddings through a fully connected network to compute action predictions for the BCL as described in Section IV-C.

V. Experiments

We evaluate the performance of our model on twenty latched doors, with 6 doors for training (3 left swinging and 3 right swinging) and 14 solely for evaluation (7 left swinging and 7 right swinging). For each door, we evaluate with 30 trials on two identical mobile manipulators: Robot A and Robot B. Note that only Robot A was used to collect training data. We evaluated models throughout the day from 9am to 5pm. Note that as these rooms are also in use by others, the types of objects and poses of interior furniture were continuously changing during our multi-week evaluations.

A. Evaluation Protocol

We use the same initial setup as during data collection and follow the same guidelines to determine task success/failure (see Section III-C.1). After initial setup, the policy controls the robot autonomously to perform the task. The safety operator can intervene at any moment to stop the robot if

Method	Total	Seen	Unseen
RGB - Real-Only (baseline)	28% ± 1.8	56% ± 3.7	14% ± 1.7
RGB - Naive (baseline)	25% ± 1.8	48% ± 3.7	15% ± 1.7
RGB - GAN (baseline)	38% ± 2.0	56% ± 3.7	31% ± 2.3
RGB - TCL	**50% ± 2.0**	**74% ± 3.3**	**40% ± 2.4**

TABLE I

DOOR OPENING SUCCESS RATE (%) ± STANDARD DEVIATION IN **REAL** OF DIFFERENT METHODS BASED ON 600 TRIALS PER EXPERIMENT.

Method	Total	Seen	Unseen
RGB - TCL	50% ± 2.0	74% ± 3.3	40% ± 2.4
Depth - TCL	**69% ± 1.9**	**81% ± 3.0**	**64% ± 2.3**
RGBD - TCL	59% ± 2.1	**80% ± 2.0**	50% ± 2.4

TABLE II

PERFORMANCE COMPARISON OF USING DIFFERENT SENSORY MODALITIES.

needed, which automatically marks the particular evaluation as a failure. All models are trained to predict task termination based on the input images. A policy which does not terminate within a timeout of two minutes is also marked as a failure.

We consider three baseline approaches: 1) RGB Real Only: trained only on the real data, 2) RGB-Naive Mixing: trained on naively mixing of sim and real images, 3) RGB-GAN [13], trained on three sources of data: RGB sim images, RGB real images, and RGB sim images adapted using a sim2real GAN. The last two baselines are ablations of our method, with 1) ablating domain adaptation entirely and 2) ablating real2sim adaptation and TCL.

We compare the baselines against an RGB instance of our method, i.e. RGB-TCL: An RGB-only model with TCL on the three variations of input images described in Section IV-B, fed from both sim and real datasets. In a separate experiment, we further compare the performance of RGB-TCL w.r.t. two other variants: Depth-TCL: Similar to RGB-TCL, but with depth images as input, and 3) RGBD - TCL: A multi-sensor variant with both RGB and depth images.

To account for variations in model training and create a fair comparison, we train three models for each approach with different random seeds and export new model checkpoints at 10 minute intervals. We use 250 simulation worker instances to evaluate the performance of each checkpoint in simulation. This thorough simulation evaluation is necessary to pick the right checkpoint; for imitation learning models, we cannot reliably determine when a model starts to overfit and then apply early stopping solely through the offline validation dataset. Based on sim evaluations across $\sim$300 checkpoints and three models, we evaluate the top-three checkpoints in a blind real-world evaluation: checkpoints are chosen at random between episodes so operators do not know which models they evaluate. Note that for the Real-Only baseline, we simply pick the last checkpoint of each model since we can no longer use simulation for checkpoint selection.

B. Results

The experiment results on latched door opening success are provided in Table I. We report estimated standard deviation for each experiment as $\sqrt{p(1-p)/(n-1)}$, assuming n trials that are i.i.d. Bernoulli variables with success rate p. As expected, RGB-Naive and Real-Only have the worst performance of 25% and 28% respectively. The former due to the lack of an explicit forcing function to reduce the domain gap, and the latter due to not having a proper tool to pick the best checkpoint. Using the RetinaGAN sim-to-real model, RGB-GAN improves 13% over the RGB-Naive model. Finally, by imposing the task consistency loss at both feature and action levels, the RGB-TCL model outperforms RGB-GAN baselines by 12%. In terms of sensory input, as summarized in Table II, TCL works well with all the three variations and not surprisingly shows that using the *depth sensor* gives us the best performance of 69% most likely due to lower visual difference between seen and unseen doors.

Figure 2 further compares sim and real performance for one run of RGB-Naive, RGB-GAN, and RGB-TCL. We observe from the figure that: (a) Sim performance fluctuates for all methods as training progresses, despite validation losses (not shown) decreasing near monotonically. As a result, always selecting the last checkpoint or basing off of validation loss is not sufficient. (b) Variance across training steps is highest for RGB-Naive and lowest for RGB-TCL. Within RGB-Naive, we hypothesize that sim and real domains are encoded as separate features and converge separately w.r.t. task success. In contrast, RGB-TCL model encodes domain invariant features and is thus more stable. We plot real world performance of the top two checkpoints for each model and measure the average sim-real performance gap for RGB-Naive, RGB-GAN, and RGB-TCL as 49.9%, 46.4% and 21.1%, respectively.

We would like to point out that each real world evaluation takes almost a full day of two operators to finish, in contrast to $\sim$10 minutes in simulation. This solidifies the importance of reliable simulation and sim-to-real transfer in guiding checkpoint selection for evaluation.

VI. CONCLUSION

In this work we presented the Task Consistency Loss (TCL), a self-supervised method for sim and real domain adaptation at the feature and action levels. Real world robotic policy evaluation for mobile manipulators can be laborious and hazardous. TCL allows us to leverage simulation to identify promising policies for real world deployment, while mitigating the reality gap. We demonstrated our method on latched door opening, a challenging mobile manipulation task using only egocentric RGB-D camera images. With only 13.5 hours of real world demonstrations and 2.7 hours of simulated demonstrations, we showed that our method improves real world performance on both seen and unseen doors, reaching 69% success. We demonstrated that using TCL reduces the gap between sim and real model evaluations by $+12$ percentage-point relative to the best baselines. This opens an opportunity to evaluate in sim to select more optimal models for real world deployment. As a future work, we will look into addressing non-visual aspects of the domain gaps between sim and real (e.g. contact forces) for imitation learning.

REFERENCES

[1] G. N. DeSouza and A. C. Kak, "Vision for mobile robot navigation: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 24, no. 2, pp. 237–267, 2002.

[2] F. Bonin-Font, A. Ortiz, and G. Oliver, "Visual navigation for mobile robots: A survey," *Journal of intelligent and robotic systems*, vol. 53, no. 3, pp. 263–296, 2008.

[3] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi, "Target-driven visual navigation in indoor scenes using deep reinforcement learning," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 3357–3364.

[4] M. Khansari, D. Kappler, J. Luo, J. Bingham, and M. Kalakrishnan, "Action image representation: Learning scalable deep grasping policies with zero real world data," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 3597–3603.

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

[6] C. Li, F. Xia, R. Martin-Martin, and S. Savarese, "Hrl4in: Hierarchical reinforcement learning for interactive navigation with mobile manipulators," 2019.

[7] C. Wang, Q. Zhang, Q. Tian, S. Li, X. Wang, D. Lane, Y. Petillot, and S. Wang, "Learning mobile manipulation through deep reinforcement learning," *Sensors*, vol. 20, no. 3, p. 939, 2020.

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

[9] T. Zhang, Z. McCarthy, O. Jow, D. Lee, X. Chen, K. Goldberg, and P. Abbeel, "Deep imitation learning for complex manipulation tasks from virtual reality teleoperation," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 5628–5635.

[10] E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn, "BC-z: Zero-shot task generalization with robotic imitation learning," in *5th Annual Conference on Robot Learning*, 2021. [Online]. Available: https://openreview.net/forum?id=8kbp23tSGYv

[11] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2017, pp. 23–30.

[12] F. Sadeghi and S. Levine, "CAD2RL: Real single-image flight without a single real image," in *Robotics: Science and Systems(RSS)*, 2017.

[13] D. Ho, K. Rao, Z. Xu, E. Jang, M. Khansari, and Y. Bai, "Retinagan: An object-aware approach to sim-to-real transfer," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 10 920–10 926.

[14] M. Stuede, K. Nuelle, S. Tappe, and T. Ortmaier, "Door opening and traversal with an industrial cartesian impedance controlled mobile robot," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 966–972.

[15] A. Gupta, A. Murali, D. Gandhi, and L. Pinto, "Robot learning in homes: Improving generalization and reducing dataset bias," *arXiv preprint arXiv:1807.07049*, 2018.

[16] C. Sun, J. Orbik, C. Devin, B. Yang, A. Gupta, G. Berseth, and S. Levine, "Fully autonomous real-world reinforcement learning for mobile manipulation," 2021.

[17] F. Xia, C. Li, R. Martín-Martín, O. Litany, A. Toshev, and S. Savarese, "Relmogen: Leveraging motion generation in reinforcement learning for mobile manipulation," 2021.

[18] A. Jain and C. C. Kemp, "Behavior-based door opening with equilibrium point control," 2009.

[19] A. Petrovskaya and A. Y. Ng, "Probabilistic mobile manipulation in dynamic environments, with application to opening doors." in *IJCAI*, 2007, pp. 2178–2184.

[20] A. J. Schmid, N. Gorges, D. Goger, and H. Worn, "Opening a door with a humanoid robot using multi-sensory tactile feedback," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 285–291.

[21] L. Peterson, D. Austin, and D. Kragic, "High-level control of a mobile manipulator for door opening," in *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000)(Cat. No. 00CH37113)*, vol. 3. IEEE, 2000, pp. 2333–2338.

[22] S. Gu, E. Holly, T. Lillicrap, and S. Levine, "Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 3389–3396.

[23] T. Welschehold, C. Dornhege, and W. Burgard, "Learning mobile manipulation actions from human demonstrations," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 3196–3201.

[24] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[25] K. Bousmalis, A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, M. Kalakrishnan, L. Downs, J. Ibarz, P. Pastor, K. Konolige, *et al.*, "Using simulation and domain adaptation to improve efficiency of deep robotic grasping," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 4243–4250.

[26] S. James, P. Wohlhart, M. Kalakrishnan, D. Kalashnikov, A. Irpan, J. Ibarz, S. Levine, R. Hadsell, and K. Bousmalis, "Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 627–12 637.

[27] K. Rao, C. Harris, A. Irpan, S. Levine, J. Ibarz, and M. Khansari, "Rl-cyclegan: Reinforcement learning aware simulation-to-real," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 157–11 166.

[28] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[29] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The journal of machine learning research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[30] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan, "Domain separation networks," *Advances in neural information processing systems*, vol. 29, pp. 343–351, 2016.

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

[32] O. J. Hénaff, A. Srinivas, J. D. Fauw, A. Razavi, C. Doersch, S. M. A. Eslami, and A. van den Oord, "Data-efficient image recognition with contrastive predictive coding," 2020.

[33] X. Chen and K. He, "Exploring simple siamese representation learning," 2020.

[34] P. Wenzel, Q. Khan, D. Cremers, and L. Leal-Taixé, "Modular vehicle control for transferring semantic information between weather conditions using gans," in *Conference on Robot Learning*. PMLR, 2018, pp. 253–269.

[35] T. Hodaň, M. Sundermeyer, B. Drost, Y. Labbé, E. Brachmann, F. Michel, C. Rother, and J. Matas, "Bop challenge 2020 on 6d object localization," in *European Conference on Computer Vision*. Springer, 2020, pp. 577–594.

[36] B. Zhou, P. Krähenbühl, and V. Koltun, "Does computer vision matter for action?" in *Science Robotics 22 May 2019: Vol. 4, Issue 30*, 2019.

[37] A. Amiranashvili, A. Dosovitskiy, V. Koltun, and T. Brox, "Motion perception in reinforcement learning with dynamic objects," in *"Conference on Robot Learning (CoRL)"*, 2019. [Online]. Available: "http://lmb.informatik.uni-freiburg.de/Publications/2018/AB18a"

[38] K. Fang, Y. Bai, S. Hinterstoisser, S. Savarese, and M. Kalakrishnan, "Multi-task domain adaptation for deep learning of instance grasping from simulation," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 3516–3523.

[39] X. Yan, J. Hsu, M. Khansari, Y. Bai, A. Pathak, A. Gupta, J. Davidson, and H. Lee, "Learning 6-dof grasping interaction via deep geometry-aware 3d representations," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 3766–3773.

[40] X. Yan, M. Khansari, J. Hsu, Y. Gong, Y. Bai, S. Pirk, and H. Lee, "Data-efficient learning for sim-to-real robotic grasping using deep point cloud prediction networks," *arXiv preprint arXiv:1906.08989*, 2019.

[41] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the fourteenth international conference on artificial intelligence and*

statistics. JMLR Workshop and Conference Proceedings, 2011, pp. 627–635.

[42] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[43] V. Tankovich, C. Hane, Y. Zhang, A. Kowdle, S. Fanello, and S. Bouaziz, "Hitnet: Hierarchical iterative tile refinement network for real-time stereo matching," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 362–14 372.

[44] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[45] P. J. Huber, "Robust Estimation of a Location Parameter," *The Annals of Mathematical Statistics*, vol. 35, pp. 73–101, 1964.