TRANSLATING ROBOT SKILLS: LEARNING UNSUPERVISED SKILL CORRESPONDENCES ACROSS ROBOTS

Anonymous authors
Paper under double-blind review

ABSTRACT

In this paper, we explore how we can endow robots with the ability to learn correspondences between its own skills, and those of morphologically different robots in different domains, in an entirely unsupervised manner. We make the insight that different morphological robots use similar task strategies to solve similar tasks. Based on this insight, we frame learning skill correspondences as a problem of matching distributions of sequences of skills across robots. We then present an unsupervised objective that encourages a learnt skill translation model to match these distributions across domains, inspired by recent advances in unsupervised machine translation. Our approach is able to learn semantically meaningful correspondences between skills across 3 robot domain pairs despite being completely unsupervised. Further, the learnt correspondences enable the transfer of task strategies across robots and domains. We present dynamic visualizations of our results at https://sites.google.com/view/translatingrobotskills/home.

1 INTRODUCTION

Humans have the ability to efficiently learn to perform tasks by watching others demonstrate similar tasks. For example, children quickly learn the skills needed to play a new sport by watching their parents perform skills such as kicking a ball. Remarkably, they are able to do learn from these visual demonstrations despite significant differences between themselves and the demonstrator, including visual perspectives, environments, kinematic and dynamic properties, and morphologies. This ability may be attributed to two factors; firstly, that humans have well developed motor skills that we can execute with little effort. Second, we are able to recognize the sequence of skills (or the high-level strategy) the demonstrator uses, and understand a corresponding set of skills we can execute ourselves.

In this paper, we explore how we can endow robots with this ability - i.e. enable robots to adopt task strategies (in terms of sequences of skills) from different robot demonstrators, and then execute corresponding skills to solve similar tasks. Skills provide a natural framework to address this problem. Sequences of skills can represent high-level task strategies; skills abstract away low-level details of the observed demonstrations and focus instead on the commonalities across demonstrators and learners - such as the task strategy.

Key to solving this problem is the ability for the robot to identify how its owns skills correspond to those of the demonstrator. Enabling robots to identify correspondences between a demonstrator robot's skills and its own skills is tremendously powerful, since it enables the following - first, this ability allows the robot to adopt the demonstrator's strategies for solving various tasks. Second, by adopting (and *adapting*) these strategies for itself, this ability enables the robot to efficiently learn to solve a variety of tasks previously outside its repertoire. Finally, this ability allows us to understand the task strategies used by various robots in a unified manner, potentially allowing robots to learn from data from a collection of other heterogeneous robots.

Before we address how we can learn skill correspondences, let us begin by defining our problem setting more formally. Given access to a set of unlabelled demonstrations of each robot solving a variety of tasks, and a method for extracting skills from those demonstrations, our goal is to learn correspondences between the skills of different robots.

Learning such correspondences is straightforward when supervised pairs of skills or trajectories are available. However, collecting and annotating such data is time-consuming and tedious, and requires

a high degree of human expertise. Moreover, it is difficult to procure such supervised data at the scale necessary to successfully learn correspondences across a diverse set of skills. In contrast, we explore whether we can learn such skill correspondences in an *unsupervised* manner, i.e. without access to supervised skill or trajectory pairs. The ability to learn unsupervised skill correspondences is powerful; it enables the use of the large scale unlabelled, unsegmented diverse demonstration datasets that have recently become available in robotics (Sharma et al., 2018; Mandlekar et al., 2018). This in turn allows for learning a broader scope of correspondences that are useful for a diverse set of tasks.

Unfortunately, learning unsupervised correspondences is a difficult problem to solve - without supervision, it is difficult to guide learning towards the right correspondences. While this may be alleviated to some extent by incorporating unsupervised constraints (Zhu et al., 2017; Ganin et al., 2016; Zhou et al., 2019), there are many spurious correspondences that satisfy these constraints.

How can one then go about learning such skill correspondences in an unsupervised manner? We propose leveraging the two following insights to do so. Our first insight is that different morphological robots use similar task strategies (in terms of sequences of skills) to solve similar tasks. Anecdotally, we observe that this is true - for example, a robot pouring out a cup of tea would likely first reach for the kettle, grasp it, move it appropriately over the cup, and then begin to pour the tea, irrespective of its exact morphology. Based on this insight, we derive an objective that guides our learning to meaningful correspondences by using similar skill sequences to solve similar tasks. Our second insight is that learning skill correspondences without access to supervised data closely mirrors unsupervised machine translation (UMT), where the objective is to learn a translation between representations in different languages (such as between word embeddings, or sentence encodings, in two languages), without access to parallel data, and instead only provided with monolingual corpora (Conneau et al., 2017; Lample et al., 2018a; Zhou et al., 2019). This insight allows us access to the machinery that UMT uses for our own problem.

We approach the problem of learning unsupervised skill correspondences by learning a translation model to map from the skill space on one "source" robot to the skill space on another "target" robot. We construct an unsupervised objective that encourages the translation model to respect our first insight - i.e. to preserve the sequences of skills observed across both the (translated source) and target robots. To do this, we take inspiration from our second insight, and specifically from Zhou et al. (2019), and construct explicit probability density models over the sequences of skills observed on the source and target robots, and then train the translation model to match these densities.

We evaluate the ability of our approach to learn unsupervised skill correspondences on three different robot pairs. Our approach is able to learn semantically meaningful correspondences across each of these 3 domain pairs, despite being completely unsupervised. We also evaluate how well the learn correspondences transfer task strategies across domains, by evaluating the transferred task startegies across a suite of downstream tasks. Our results are visualized at https://sites.google.com/view/translatingrobotskills/home.

2 Related Work

Skill Learning: Eysenbach et al. (2019); Sharma et al. (2020) both address unsuperivsed skill learning from interaction data, by constructing information theoretic approaches. Fox et al. (2017); Krishnan et al. (2017); Shankar et al. (2020); Sharma et al. (2018); Shankar & Gupta (2020); Kipf et al. (2019) instead learn skills from unlabelled demonstration data by performing latent skill inference. Gregor et al. (2019); Kim et al. (2019) address learning more generic temporal abstractions. While these skill learning frameworks consider skills in the context of a single domain, they broadly form the basis for our work. In particular, we make use of Shankar & Gupta (2020)'s formulation to extract skills from demonstrations, but believe our work could be used to learn skill (and more generally, abstraction) correspondences with a wide variety of these methods.

Unsupervised Correspondence Learning: As noted in the introduction, our problem bears a close resemblance with unsupervised machine translation (Conneau et al., 2017; Zhou et al., 2019), and unpaired image translation (Zhu et al., 2017; Park et al., 2020). Specifically, they all share the notion of learning correspondences across representations learnt from unpaired, "monolingual" data. We briefly describe related work in each of these areas below, before considering their applicability to our problem.

- **Unsupervised Correspondence in Machine Translation:** The unsuperivsed machine translation problem has been widely studied by the language community. Notably, Conneau et al. (2017) leverage domain-adversarial training (Ganin et al., 2016) to align word embeddings of two langauges. Sennrich et al. (2016) use the idea of back-translation to constraint the learnt translation models across languages. Lample et al. (2018a) couple back-translation with a denoising autoencoder to ensure robustness to noisy translations. Zhou et al. (2019) learn bilingual word embeddings by constructing explicit density functions over the word embedding space, and then matching these densities using normalizing flows.

- **Unsupervised Correspondence in Image Translation:** The vision community has similarly explored the unpaired *image-to-image* translation setting Zhu et al. (2017); Park et al. (2020). Zhu et al. (2017) combined cycle-consistency with adversarial training to achieve impressive photorealistic results. Park et al. (2020) replaced the cycle-consistency objectives by introducing a contrastive loss to learn a translation function instead.

- **Unsupervised Correspondence in the Video Domain:** The problem of learning unsupervised correspondences has applications in the video domain as well, in form of video retargeting (Bansal et al., 2018), or object correspondence within a video (Wang et al., 2019). Bansal et al. (2018) extended Cycle-GAN Zhu et al. (2017) to video retargeting by adding a temporal consistency loss. Wang et al. (2019) derive a cycle-consistency objective, where the cycle is defined *through time*, rather than purely across domains.

These approaches have all enjoyed notable success in their respective settings, but unfortunately are not trivially applicable to our problem setting, owing to the following reasons. The domain-adversarial training (Ganin et al., 2016) underlying several of these approaches (Conneau et al., 2017; Lample et al., 2018a;b; Zhu et al., 2017; Bansal et al., 2018) is notoriously difficult to train. It is also prone to the mode dropping problem (Li & Malik, 2019). Even with constraints such as cycle-consistency (Zhu et al., 2017; Sennrich et al., 2016) which prevent this, adversarial training requires strong constraints such as the pixel-wise identity losses used in Zhu et al. (2017). Further, approaches such as Conneau et al. (2017); Zhou et al. (2019) rely heavily on the inherent similarities of the spaces to be aligned (word embeddings across languages often possess remarkable similarity, and are often simply linear transforms of one another). In general learnt skill representations do not possess this property.

Shortcomings nonwithstanding, these approaches provide valuable insight into how we may go about learning unsupervised correspondences between robot skills. In particular, we take inspiration from Zhou et al. (2019), and opt for an explicit density model based approach over an implicit adversarial one (Ganin et al., 2016), since it avoids the mode-dropping problem, and is notably easier and more stable to train.

Domain Transfer in Robotics: The robotics community has taken interest in cross-domain transfer of policies in recent years (Sermanet et al.; Gupta et al., 2017; Ammar et al., 2015; Raimalwala et al., 2016; Zhang et al., 2021; Kim et al., 2020; Hejna et al., 2020; Devin et al., 2017) . Gupta et al. (2017); Sermanet et al. address morphological transfer in robotics by learning feature spaces invariant to morphology and viewpoints respectively, but both require paired data to do so. Hejna et al. (2020); Devin et al. (2017) address morphological transfer by modularity in their policies. Hejna et al. (2020) adopt modularity in a hierarchical sense, but leverage common grounding of subgoals across morphologies to perform the transfer. In contrast, we do not assume access to any such known common grounding. Liu et al. (2020) addresses cross-morphology transfer by state aware imitation learning, as does Schroecker & Isbell (2017). Similarly, Ammar et al. (2015) address cross task transfer via manifold alignment, but also only do so in state space. Zhang et al. (2021); Kim et al. (2020) both address learning low-level state and action correspondences from unpaired and unaligned interaction and demonstration data respectively. While similar in spirit, our work argues for learning high-level skill correspondence instead. Since skills abstract away low-level details that may differ across robots, transferring task strategies via skills rather than state-actions is more natural and powerful choice. For example, skills such as reaching and placing are well grounded across robot manipulators with different degrees of freedom, while there is no obvious correspondence in the low-level actions they execute or states they visit. Further, Zhang et al. (2021); Kim et al. (2020) both adopt generative-adversarial training approaches. As noted above, such methods are notoriously difficult to train, and use hand-crafted auxilliary losses to bias solutions towards correct correspondences, such as the identity loss used in Zhang et al. (2021). In contrast, we adopt an

explicit density model based alignment, bypassing the need for both unstable adversarial training, as well as auxilliary losses altogether.

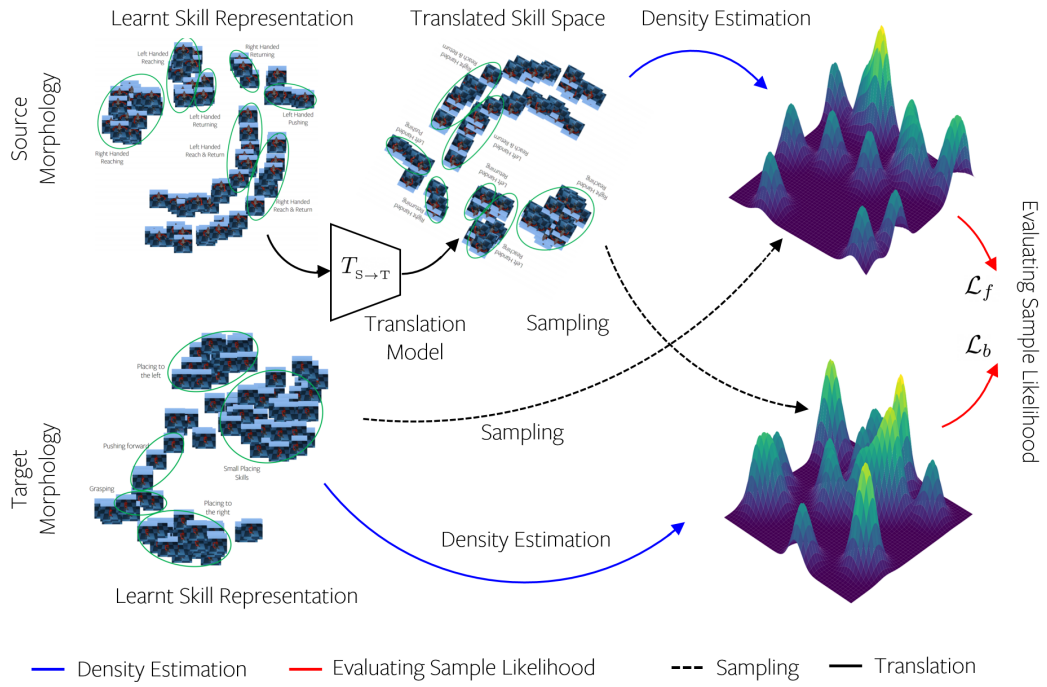

Figure 1: Overview of our approach. We translate the original learnt skill space to the target domain. We then construct explicit density estimates over the original target and translated source skill-tuple spaces. We then maximize the likelihood of randomly sampled translated source and original target skill-tuples under these densities respectively.

3 Approach

TO DO: DESCRIBE BIG IDEAS HERE.. / introduction for section.

3.1 Pre-requisites

We begin describing our approach by first reviewing an important prerequisite - the skill learning pipeline. Shankar & Gupta (2020) derive a method to learn robotic skills from unlabelled robotic demonstrations, that serves as the backbone of our work; we briefly describe it below. They first represent robotic skills as continuous latent variables z, and introduce a version of variational inference to infer these skills or latent variables. They train a variational encoder $q(z|\tau)$ that takes as input a robot trajectory $\tau = \{s_1, a_1, ...s_{n-1}, a_{n-1}, s_n\}$, and outputs a sequence of skill encodings $z = \{z_1, z_2, ...z_n\}$. They also train a latent conditioned policy $\pi(a|s, z)$ that takes as input robot state s, and the chosen skill encoding z, and predicts the low-level action a that the robot should execute. We direct the reader to their paper for a more thorough description of their skill learning approach. While our work is specifically implemented with the skill-learning framework from Shankar & Gupta (2020), we believe other skill learning techniques (Eysenbach et al., 2019; Sharma et al., 2020; Shankar et al., 2020; Kipf et al., 2019) could be used as well. **TO DO: REFER TO THIS AS TVI**

3.2 Problem Setting

Consider two robots with potentially differing morphologies and environments (or "domains"). We represent these domains as source and target domains, or M_{S} and M_{T} respectively. Associated with each domain is an unlabelled, and unsegmented dataset of demonstrations of the robot performing various tasks; we represent these datasets for the two domains as D_{S} and D_{T} respectively. We consider that D_{S} and D_{T} are collected independently, on a potentially intersecting (but not identical) set of tasks. This is equivalent to the setting in machine translation without *parallel* data, with access only to *monolingual* corpora.

We also assume access to a skill-learning pipeline, that can take in such an unlabelled demonstration dataset D of a robot M performing various tasks, and learns a representation $\mathbb{Z}$ of skills on a given robot. We specifically use the Temporal Variational Inference formulation of Shankar & Gupta (2020), but we believe our approach is compatible with any unsupervised skill-learning approach that affords a continuous representation space.

Here, we train the skill-learning pipeline independently on each of the domain datasets D_{S} and D_{T}. This affords us skill encoders q_{S} and q_{T}, and latent conditioned policies π_{S} and π_{T} for each domain. We may then encode trajectories in both domains $\tau_{\text{S}} \in D_{\text{S}}$ and $\tau_{\text{T}} \in D_{\text{T}}$ into their constituent skills $\{z_{\text{S}}^t\}_{t=1}^{|\tau_{\text{S}}|}$ and $\{z_{\text{T}}^t\}_{t=1}^{|\tau_{\text{T}}|}$, via q_{S} and q_{T} respectively. We represent the space of skills learnt on the entire dataset in each of the two domains as $\mathbb{Z}_{\text{S}}$ and $\mathbb{Z}_{\text{T}}$ respectively.

Our goal is to learn semantically meaningful correspondences between skills in the source domain, $z_{\text{S}} \in \mathbb{Z}_{\text{S}}$, and those in the target domain, $z_{\text{T}} \in \mathbb{Z}_{\text{T}}$, that helps solve tasks in the target domain M_{T}. We specify these correspondences by learning a "translation" function $T_{\text{S}\to\text{T}} : \mathbb{Z}_{\text{S}} \to \mathbb{Z}_{\text{T}}$, that maps skills in the source domain z_{S}, to translated skills in the target domain $z_{\text{S}\to\text{T}}$. Learning meaningful correspondences thus amounts to learning a good translation model T. We can also translate the entire source space $\mathbb{Z}_{\text{S}}$ to the target domain, to retrieve the translated source space, $\mathbb{Z}_{\text{S}\to\text{T}}$. This mirrors the word translation models present in Conneau et al. (2017); Zhou et al. (2019).

3.3 Using sequential information of skills

Our first insight was that different morphological robots follow similar strategies to address similar tasks - in other words, sequences of skills executed by two different robots ought to belong to similar distributions. We note that it is important to consider *sequences* of skills here rather than skills in isolation, since the ordering of skills gives us additional information to bias our learning towards the right correspondences. Constraining learning based on individual skills alone could lead to erroneous correspondences. For example, learning that "reaching" skills in one domain correspond to "grasping" skills in the other domain is evidently incorrect, but is an error that sequential information could bias learning away from, since grasping skills are systematically executed *after* reaching skills.

While ideal in principle, processing entire sequences of skills is computationally infeasible for arbitrarily long trajectories. Instead, we draw inspiration from the simple yet powerful bi-gram models in the NLP community, and consider 1-step transitions between skills, or consecutive *tuples* of observed skills. While less expressive than the full sequence of skills, these tuples retain enough information of the ordering of skills to guide the correspondence learning towards the right solution, and have the advantage of being far more computationally tractable. We thus construct a skill-tuple space, which represents the various transitions between skills observed in a given domain. Each consecutive pair of skills (z^t, z^{t+1}) in the original space $\mathbb{Z}$ of a given domain is represented as a single point x^t in the skill-tuple space $\mathbb{X}$ of that domain. For initial and terminal skills $z^{t=0}$ and $z^{t=|\tau|}$, we pad these tuples, and treat the preceeding and succeeding skill encodings as appropriately dimensioned 0's respectively. The skill-tuple spaces for both source and target domains are represented as $\mathbb{X}_{\text{S}}$ and $\mathbb{X}_{\text{T}}$ respectively. Each of these spaces are implemented as a set of skill-tuples from each respective domain, i.e. $\mathbb{X}_{\text{S}} = \{x_{\text{S},i}\}_{i=1}^{N_{\text{S}}}, \mathbb{X}_{\text{T}} = \{x_{\text{T},i}\}_{i=1}^{N_{\text{T}}}$. The procedure to construct these spaces, presented in the sub-routine of algorithm 1, runs TVI, and then simply assembles skill-tuples from the resultant skills. As before, we also construct a translated skill-tuple space, $\mathbb{X}_{\text{S}\to\text{T}}$, that is simply the translation of all skills present in $\mathbb{X}_S$, i.e. $\mathbb{X}_{\text{S}\to\text{T}} = \{x_{S\to T,i}\}_{i=1}^{N_S}$.

Translating a skill-tuple $x_{\text{S}}^t = (z_{\text{S}}^t, z_{\text{S}}^{t+1})$ from the source domain to the target domain is done by translating the individual skills in the tuple, z_{S}^t and z_{S}^{t+1}, to the target domain, $z_{\text{S}\to\text{T}}^t$ and $z_{\text{S}\to\text{T}}^{t+1}$, and constructing the translated target domain skill-tuple as $x_{\text{S}\to\text{T}}^t = (z_{\text{S}\to\text{T}}^t, z_{\text{S}\to\text{T}}^{t+1})$.

3.4 Distributional perspective on learning translation functions

Ideally, we would like our translation model to satisfy two properties. The first is that it ought to translate source skills z_{S} in such a way that they belong to the distribution of target skills. The second is that it ought to capture all modes of skills that exist in the target domain, since dropping these modes limits the set of skills the target robot can use, reducing its capabilities.

Generative-adversarial (Goodfellow et al., 2014) and domain-adversarial training approaches (Ganin et al., 2016) provides a natural way to satisfy the first property, since they optimize solely for how

realistic the generated samples are (or how indistinguishable source and target domain features are (Ganin et al., 2016)). However, they do not optimize for capturing all the modes of the "real" data, and hence suffer from the mode-dropping problem (Li & Malik, 2019), making them ill-suited with respect to the second mode-covering property.

Instead, we choose to approach this problem from an *explicit* distribution perspective, rather than implicitly, as in GANs. We maintain explicit probability density estimates over the translated source and target skill-tuple spaces, $\mathbb{X}_{\mathrm{S}\to\mathrm{T}}$ and $\mathbb{X}_{\mathrm{T}}$. This allows us to train the translation model to satisfy both desired properties simply by maximizing likelihoods under these two distributions, as we elucidate on below.

Consider constructing an explicit probability density function $p_{\mathrm{T}}(x)$, over the target domain skill-tuple space $\mathbb{X}_{\mathrm{T}}$. We follow Zhou et al. (2019)'s choice of representing $p_{\mathrm{T}}(x)$ using a Gaussian Mixture Model (GMM), owing to its expressive power and efficiency in low-dimensional spaces, but note that a wide variety of explicit density estimators may be used here. Here, $p_{\mathrm{T}}(x)$ is a GMM with N_{T} Gaussian kernels, each centered at a target skill-tuple $x'_{\mathrm{T},i} \sim \mathbb{X}_{\mathrm{T}}$:

$$p_{\mathrm{T}}(x) = \sum_{i=1}^{N_{\mathrm{T}}} p(x'_{\mathrm{T},i})\, p(x|x'_{\mathrm{T},i}) = \sum_{i=1}^{N_{\mathrm{T}}} p(x|x'_{\mathrm{T},i}) \tag{1}$$

Zhou et al. (2019) use kernel weights $p(x'_{\mathrm{T},i})$ proportional to the frequency of the word the kernel represents. In our setting, each kernel is simply one of the skill-tuples in the *continuous* target skill space $\mathbb{X}_{\mathrm{T}}$, motivating our choice of equal mixture weights over all kernels (i.e. skill-tuples), i.e. $p(x'_{\mathrm{T},i}) = 1/N_{\mathrm{T}}$. The mixture components $p(x|x'_{\mathrm{T},i})$ are specified by a fixed variance Gaussian centered at $x'_{\mathrm{T},i}$, i.e. $p(x|x'_{\mathrm{T},i}) = \mathcal{N}(x|x'_{\mathrm{T},i}, \sigma^2)$, where σ specifies the standard deviation of the Gaussian kernel.

This explicit density estimate affords us the ability to achieve the first property. We can train a translation model to translate skills in such a way that the translated skill tuples $x_{\mathrm{S}\to\mathrm{T}}$ belong to the distribution of target skill-tuples $p_{\mathrm{T}}(x)$. We do this by maximizing the likelihood of the translated skill-tuples $x_{\mathrm{S}\to\mathrm{T}}$, under the target density $p_{\mathrm{T}}(x)$. We call this objective the forward objective, $\mathcal{L}_f$:

$$\mathcal{L}_f = \mathbb{E}_{x_{\mathrm{S}}\sim p(x_{\mathrm{S}}), x_{\mathrm{S}\to\mathrm{T}} \sim T_{\mathrm{S}\to\mathrm{T}}(.|x_{\mathrm{S}})} \Big[\log p_{\mathrm{T}}(x_{\mathrm{S}\to\mathrm{T}}) \Big] \tag{2}$$

To encourage the second mode-covering property in the translation model, we can also optimize the likelihood of the *target* skill-tuples under the *translated source* skill-tuple space. To do so, we first construct an explicit probability density function $p_{\mathrm{S}\to\mathrm{T}}(x)$ over the translated source skill-tuple space, $\mathbb{X}_{\mathrm{S}\to\mathrm{T}}$, that is equivalent to $p_{\mathrm{T}}(x)$. $p_{\mathrm{S}\to\mathrm{T}}(x)$ is also similarly represented as a GMM, with $N_{\mathrm{S}\to\mathrm{T}}$ equally weighted Gaussian kernels, each centered at a *translated* skill-tuple $x'_{\mathrm{S}\to\mathrm{T},i} \sim \mathbb{X}_{\mathrm{S}\to\mathrm{T}}$:

$$p_{\mathrm{S}\to\mathrm{T}}(x) = \sum_{i=1}^{N_{\mathrm{S}\to\mathrm{T}}} p(x'_{\mathrm{S}\to\mathrm{T},i})\, p(x|x'_{\mathrm{S}\to\mathrm{T},i}) = \sum_{i=1}^{N_{\mathrm{S}\to\mathrm{T}}} p(x|x'_{\mathrm{S}\to\mathrm{T},i}) \tag{3}$$

As before, the conditional $p(x|x'_{\mathrm{S}\to\mathrm{T},i}) = \mathcal{N}(x|x'_{\mathrm{S}\to\mathrm{T},i}, \sigma^2)$. We may then optimize the likelihood of a set of target skill-tuples x_{T} under $p_{\mathrm{S}\to\mathrm{T}}(x)$. We call this the *backward* objective $\mathcal{L}_b$:

$$\mathcal{L}_b = \mathbb{E}_{x_{\mathrm{T}}\sim p(x_{\mathrm{T}})} \Big[\log p_{\mathrm{S}\to\mathrm{T}}(x_{\mathrm{T}}) \Big] \tag{4}$$

By combining these two objectives we may construct our final objective for training the translation model $\mathcal{L}$, by simply combining these two objectives:

$$\mathcal{L} = \mathcal{L}_f + \mathcal{L}_b = \mathbb{E}_{x_{\mathrm{S}}\sim p(x_{\mathrm{S}}), x_{\mathrm{S}\to\mathrm{T}} \sim T_{\mathrm{S}\to\mathrm{T}}(.|x_{\mathrm{S}})} \Big[\log p_{\mathrm{T}}(x_{\mathrm{S}\to\mathrm{T}}) \Big] + \mathbb{E}_{x_{\mathrm{T}}\sim p(x_{\mathrm{T}})} \Big[\log p_{\mathrm{S}\to\mathrm{T}}(x_{\mathrm{T}}) \Big] \tag{5}$$

Parsing the objective: The explicit density based objective we present to train our translation model has several useful features that we note here. In contrast with adversarial training methods, which require alternating training phases, many stability tricks (such as gradient penalties, etc.), our training method amounts to simple maximum likelihood (albeit in two directions). It is hence simpler, more stable, and quicker to converge. Being fully differentiable, it can easily be optimized by gradient

Algorithm 1 Translating Robot Skills

Input: $D_{\text{S}}, D_{\text{T}}, q_{\text{S}}, q_{\text{T}}, \pi_{\text{S}}, \pi_{\text{T}}$
▷ Require demonstration datasets, trained skill encoders & decoders for source & target domains
Output: $T_{\text{S}\to\text{T}}(\,.\,|z_{\text{S}})$ ▷ Output translation model
1: Initialize $T_{\text{S}\to\text{T}}(\,.\,|z_{\text{S}})$ ▷ Initialize translation model
2: $\mathbb{X}_{\text{S}} \leftarrow$ BUILDSKILLTUPLESPACE(Source) ▷ Construct source skill-tuple space
3: $\mathbb{X}_{\text{T}} \leftarrow$ BUILDSKILLTUPLESPACE(Target) ▷ Construct target skill-tuple space
4: $p_{\text{T}}(x) \leftarrow \sum_{i=1}^{N_{\text{T}}} p(\,.\,|x'_{\text{T},i})$ ▷ Update target GMM density via eq. (1)
5: **for** $i \in [1, 2, ..., N_{\text{iterations}}]$ **do**
6: $\mathbb{X}_{\text{S}\to\text{T}} \leftarrow T_{\text{S}\to\text{T}}(\,.\,|\mathbb{X}_{\text{S}})$ ▷ Update translated-source skill-tuple space
7: $p_{\text{S}\to\text{T}}(x) \leftarrow \sum_{i=1}^{N_{\text{S}}} p(\,.\,|x'_{\text{S}\to\text{T},i})$ ▷ Update translated-source GMM density via eq. (3)
8: $x_{\text{S}} \sim \mathbb{X}_{\text{S}}, x_{\text{T}} \sim \mathbb{X}_{\text{T}}$ ▷ Get batch of source and target skill-tuples
9: $x_{\text{S}\to\text{T}} \sim T_{\text{S}\to\text{T}}(\,.\,|x_{\text{S}})$ ▷ Translate source skill-tuple to target domain
10: $\mathcal{L}_f \leftarrow \log p_{\text{T}}(x_{\text{S}\to\text{T}})$ ▷ Evaluate forward objective
11: $\mathcal{L}_b \leftarrow \log p_{\text{S}\to\text{T}}(x_{\text{T}})$ ▷ Evaluate backward objective
12: $\mathcal{L} \leftarrow \mathcal{L}_f + \mathcal{L}_b$ ▷ Evaluate full objective
13: Update $T_{\text{S}\to\text{T}}$ via $\nabla\mathcal{L}$ ▷ Update translation model with gradient ascent

Sub-routine: Build Skill-Tuple Space

Input: Domain M
Output: Skill-Tuple space $\mathbb{X}_M$
14: **function** BUILDSKILLTUPLESPACE(Domain M)
15: $\mathbb{X}_M \leftarrow \{\}$ ▷ Initialize empty skill-tuple space
16: **for** $i \in [1, 2, ..., N_{\text{M}}]$ **do**
17: $\tau_{\text{M},i} \sim D_{\text{M}}$ ▷ Retrieve batch of trajectories from domain dataset
18: $\{z_{\text{M}}^t\}_{t=1}^{|\tau|} \sim q_{\text{M}}(\,.\,|\tau_{\text{M},i})$ ▷ Encode trajectories as sequence of skills via domain encoder
19: $\{x_{\text{M}}^t\}_{t=1}^{|\tau|-1} \leftarrow \{(z_{\text{M}}^t, z_{\text{M}}^{t+1})\}_{t=1}^{|\tau|-1}$ ▷ Assemble skill tuples
20: $\mathbb{X}_M \leftarrow \mathbb{X}_M \cup \{x_{\text{M}}^t\}_{t=1}^{|\tau|-1}$ ▷ Add skill-tuples to skill-tuple space
21: **return** $\mathbb{X}_M$

based optimization. The bidirectional nature of our objective may be thought of as follows. $\mathcal{L}_f$ tries to push every translated skill-tuple $\forall\, x_{\text{S}\to\text{T}} \in \mathbb{X}_{\text{S}\to\text{T}}$ close to some target domain skill-tuple x_{T} (i.e., it is mode-seeking), while $\mathcal{L}_b$ tries to pull a translated skill-tuple $x_{\text{S}\to\text{T}}$ towards every target skill-tuple $\forall\, x_{\text{T}} \in \mathbb{X}_{\text{T}}$ (i.e., it is mode-covering).

Algorithm: Armed with this understanding of our objective, we now present our full algorithm for learning unsupervised correspondences between skill spaces from different robots in algorithm algorithm 1. As mentioned in section 3.3, the sub-routine BUILDSKILLTUPLESPACE runs the variational inference of Shankar & Gupta (2020), and then assembles skill-tuples. The skill-spaces in both domains are kept fixed, and so only need to be computed once initially for each domain (in steps 2 & 3), as is the case for the target density $p_{\text{T}}(x)$.

During each training iteration, we must first recompute the translation of the source skill-tuple space, since the translation model changes at every iteration. As a result, the translated-source density $p_{\text{S}\to\text{T}}$ must also be recomputed every iteration, as per eq. (3). We may then compute the forward and backward components of our objective $\mathcal{L}_f$ and $\mathcal{L}_b$, as per eq. (2) and eq. (4) stochastically over a batch of skill-tuples. Finally, we update our translation model via stochastic gradient ascent on $\mathcal{L}$.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

We first describe the domain-pairs that we learn correspondences across, along with the datasets we use to learn skills in each of these domains.

Robots: We evaluate our approach's ability to learn skills correspondences across the following robots - the Sawyer robot, the left-hand of the Baxter robot, and the right-hand of the Baxter robot. For each non-identical pair of these 3 robots, we may consider each robot as either the source and target domain, since we learn a translation function in one-direction alone. We consider 3 of the possible 6 unique domain pairs, which we list below (along with how we refer to them in our experiments) -

1. Baxter Right-Hand to Baxter Left-Hand (Bax-R to Bax-L)
2. Baxter Left-Hand to Sawyer (Bax-L to Saw)
3. Baxter Right-Hand to Sawyer (Bax-R to Saw)

Our choices of robots are dictated by the availability of large scale demonstration datasets for these respective robots.

Datasets: For the Sawyer robot, we make use of the Roboturk dataset (Mandlekar et al., 2018), which consists of roughly 2000 demonstrations across 8 different tasks. For the Baxter robot, we utilize the MIME dataset (Sharma et al., 2018). We partition the MIME dataset into two disjoint sets with solely left-handed and right-handed trajectories respectively. Each single-handed dataset also roughly contains 2000 demonstrations across 16 tasks. We treat each single-handed dataset as the respective dataset for the Baxter left-hand robot and the Baxter right-hand robot.

Skill Representation: We learn skill-representations for each of the 3 robots independently from their respective datasets, using the approach from Shankar & Gupta (2020). We use a 16-dimensional latent space for each robot to parameterize the skills, that is learnt from robot joint-states and joint velocities. We follow the preprocessing steps and training parameters specified by (Shankar & Gupta, 2020). In order to evaluate our approach, we also manually annotate 50 skills in each domain with semantic labels of what type of skills they were. We emphasize that our approach is not trained with these models, but rather is only evaluated against them.

4.2 Learning meaningful skill correspondences

The first question we would like to answer is - *"Can our unsupervised method learn meaningful skill-correspondences between skill spaces?"* We first present qualitative results towards this.

Visualizing & Translating Individual Skills: We first visualize individual skills in the source domain, by selecting a representative set of skills z_{S}, and rolling out π_{S} conditioned on z_{S}. We then translate these skills z_{S} to the target domain, $z_{\mathrm{S}\to\mathrm{T}}$, and visualize the rollout of the *target* domain policy π_{T} conditioned on $z_{\mathrm{S}\to\mathrm{T}}$. We present dynamic visualizations of these individual source domain skills and their corresponding translations in the target domain in the Translating Individual Skills section of https://sites.google.com/view/translatingrobotskills/home.

Visualizing & Translating Entire Trajectories: We also ascertain how well we can translate entire trajectories (or entire sequences of skills) across domains. One may think of this as a motion retargeting problem, or a method to transfer task strategies (as sequences of skills) across domains. We first visualize a set of source domain trajectories τ_{S}, obtained by rolling out their skill encodings $\{z_{\mathrm{S}}^t\}_{t=1}^{|\tau|}$ via the source domain policy π_{S}. We then translate these sequences of skill encodings to the target domain, i.e. $\{z_{\mathrm{S}\to\mathrm{T}}^t\}_{t=1}^{|\tau|}$, and visualize rollouts of the *target* policy π_{T} conditioned on these translated skills. We present these results in the Translating Entire Trajectories section of https://sites.google.com/view/translatingrobotskills/home.

Analysing Qualitative Skill Translations: Upon analysing the translated skills and entire trajectories, we observe the following. Despite the translation model being learned in a purely unsupervised manner, without access to labels of any kind, the model is able to learn appreciable *coarse* semantic correspondences between skills in the source domain and the target domain across all 3 robot domain pairs. For example, source domain reaching skills correspond to target domain reaching skills across all 3 domain pairs, as do placing and pushing / sliding skills to the left or to the right, and returning skills to their rest configuration. While the model also often translates finer skills (such as twisting or grasping skills) appropriately across domains, it does not always learn perfect *finer* correspondences. Given our method exploits similar contexts of skills across domains to learn correspondences, these results are understandable, since such finer skills are often executed in similar contexts, such as after reaching or before placing skills.

Table 1: Evaluating Skill-Tuple Distribution Matching via our approach against various baselines, across all domain pairs. Baselines are adapted from (1): Ganin et al. (2016), (2): Zhu et al. (2017), (3): Liu et al. (2020). Lower is better for all metrics except the label accuracy. * - Running additional experiments that may change these numbers.

Domain Pair	Approach:	$\mathcal{L}_f$	$\mathcal{L}_b$	Chamfer Distance	Cycle Error	Label Accuracy	Supervised Error
Bax-R to Bax-L	DomAd [1]	38.1	43.4	20.1*	17.4*	10.3%	12.2*
	Cycle GAN [2]	32.1	39.8	17.0*	9.2*	14.2%	12.9*
	Ours	**14.3**	**20.1**	**8.2**	10.2*	62.8%*	8.1*
Bax-L to Saw	DomAd [1]	50.9	60.1	24.2*	21.3*	9.8%*	14.9*
	Cycle GAN [2]	38.5	53.7	17.1*	8.8*	12%*	15.4*
	Ours	**16.8**	**33.9**	**7.8**	9.0*	58.1%*	9.0*
Bax-R to Saw	DomAd [1]	54.8	80.2	21.2*	24.8*	11.1%*	19.1*
	Cycle GAN [2]	47.3	78.5	16.7*	10.3*	16.8%*	20.1*
	Ours	**15.4**	**36.5**	**7.2**	10.1*	64.6%*	12.1*

Another perspective on this is the model learns correspondences between different clusters of skills that emerge in the original skill spaces, but does not learn intra-cluster correspondences that disambiguate between different types of skills within a cluster. For example, while the model correctly translates a reaching skill on the Baxter right hand to a reaching skill on the Baxter left hand, the shape that the arm takes on during these reaching skills does not match perfectly - the right hand reaches with a bent elbow and wrist, but the left hand does not. This is because there are a number of different arm shapes observed during the various reaching skills in each domain, each occurring in similar contexts.

Upon analysing the translation of entire trajectories, we observe that despite the coarse nature of the correspondences learnt by our model, we are able to transfer the source domain trajectory's overall structure, and sequence of skills executed, remarkably well to the target domain. Our model does so despite variations in the precise shape of the robot arms, or start and end positions of these respective skills. This provides further evidence that the learnt correspondences are indeed semantically meaningful, and additionally suggests these learnt correspondences would be useful in transferring *task strategies* across domains. We believe this is a powerful result, especially so considering approach does so in a completely unsupervised manner.

Quantitative Evaluation of Skill-Tuple Distribution Matching: To quantitatively measure how well our approach can matching the distributions of skill-tuples across domains, we evaluate the following unsupervised metrics:

1. *Forward and Backward GMM densities* $\mathcal{L}_f$ and $\mathcal{L}_b$: $\mathcal{L}_f$ represents how "realistic" each translated skill-tuple looks with respect to the target skill-tuple distribution, while $\mathcal{L}_b$ represents how well the translated skills cover the modes of the target skill-tuple space.
2. *Chamfer distance:* We compute the Chamfer distance (Fan et al., 2017) between the skill-tuple spaces, which represents the nearest neighbor distances across two point sets.
3. *Cycle-reconstruction error:* We evaluate the cycle-reconstruction error (Zhu et al., 2017) between a skill z_{S}, and its translation to the target domain and back, $z_{\mathrm{S}\to\mathrm{T}\to\mathrm{S}}$.
4. *Label transfer accuracy:* We also evaluate how well the semantic labels associated with the translated annotated skill set align with the labelled target domain skills.

We compare our approach against the following alignment baselines.

1. Domain-Adversarial training: We follow Ganin et al. (2016), and match skill distributions by training the translation model to fool a discriminator network trained to identify which domain a given skill came from.
2. Cycle-GAN: We follow Zhu et al. (2017), and train two translation models between skills from source and target domains to be cycle-consistent with one another, while also optimizing a domain-adversarial style objective (Ganin et al., 2016), as in Zhu et al. (2017).

Table 2: * Evaluating Task-Strategy Transfer on Downstream Tasks: Evaluating average rewards obtained via our approach with and without finetuning against a hierarchical RL baseline, adapted from Kulkarni et al. (2016). Note that in the source domains, results are shared across approaches, and are trained via Kulkarni et al. (2016).

Domain Pair	Approach:	Downstream Task					
		Reach (Source)	Reach (Target)	Push (Source)	Push (Target)	T3 (Source)	T3 (Target)
Bax-R to Bax-L	HRL [1]		15.2		12.6		10.7
	Ours (Zero Shot)	16.4	19.2	30.1	17.2	21.3	13.4
	Ours (Fine-tune)		20.1		19.1		18.9
Bax-L to Saw	HRL [1]		13.2		11.3		17.2
	Ours (Zero Shot)	14.6	18.6	33.4	37.5	20.4	24.6
	Ours (Fine-tune)		19.1		34.1		38.1
Bax-R to Saw	HRL [1]		15.2		12.6		10.7
	Ours (Zero Shot)	16.4	20.8	30.1	28.5	21.3	29.1
	Ours (Fine-tune)		31.9		32.5		30.5

Analysis of Qualitative Results of Skill-Tuple Distribution Matching: * From table 1, we observe that our method is able to achieve a much higher semantic label transfer accuracy than the baseline approaches. Both the Cycle-GAN baseline and the domain adversarial training baselines fail to learn correct correspondences. For example, the domain adversarial baseline ends up mapping several different types of skills in the target domain to single modes of skills in the target domain, across all 3 domain pairs. The Cycle-GAN baseline somewhat mitigates this approach, but takes a shortcut in the learning and simply places a single source skill nearby every mode of target skill. In contrast, the backward loss explicitly optimizes for mode coverage of the target domain, it is able to match the skill-tuple distributions very well.

4.3 Transferring task strategies across domains

The second question we would like to answer is *"How well do the learnt correspondences help transfer task strategies across domains?"* We first retrieve task strategies (specified by sequences of skills) in the source domain across a suite of tasks adapted from Mandlekar et al. (2018), by training a high level policy on these tasks. We then translate these skills to the target domain, and then evaluate how well the translated sequence of skills performs on the same tasks in the *target* domain. We evaluate this both with and without fine-tuning the translated sequence of skills in the target domain. Without fine-tuning these translated skills, one may think of this as a zero-shot task strategy transfer. For comparison we also evaluate a policy trained in the target domain itself, as a baseline. We report the results of this transfer in (TABLE), and provide visualizations of the rollouts from the original and translated strategies in https://sites.google.com/view/translatingrobotskills/home.

Analysing results on task strategy transfer: * From table 2, we observe that our approach is able to learn correspondences that do indeed facilitate transferring task strategies across domains. Zero-shot transfer via our approach helps eliminate the need for training traditionally sample inefficient hierarchical RL (Kulkarni et al., 2016) on the target domain robot. Fine-tuning the transferred strategies via hierarchical RL (Kulkarni et al., 2016) only provides a marginal performance improvement over the transferred strategy without fine-tuning. This suggests that transferring a task strategy via our learnt correspondences is successful.

To Do: DISCUSSION / LIMITATIONS

5 Conclusion

We introduce a purely unsupervised approach to learn skill correspondences across different morphological robots from different domains. Our approach learns semantically meaningful correspondences across 3 robot domain pairs, and helps transfer task-strategies across domains, without the need for any fine-tuning, despite being completely unsupervised. We believe our approach could enable learning of correspondences across more general temporal abstractions, such as between skills and language instructions, or between skills and human video demonstrations. We hope this work helps

enable robots learn from data from heterogenous robots in different domains, and helps further the state of robot learning from demonstration.

REFERENCES

Haitham Bou Ammar, Eric Eaton, Paul Ruvolo, and Matthew E. Taylor. Unsupervised cross-domain transfer in policy gradient reinforcement learning via manifold alignment. pp. 2504–2510, 2015.

Aayush Bansal, Shugao Ma, Deva Ramanan, and Yaser Sheikh. Recycle-gan: Unsupervised video retargeting. 2018.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. *CoRR*, abs/1710.04087, 2017. URL http://arxiv.org/abs/1710.04087.

Coline Devin, Abhishek Gupta, Trevor Darrell, Pieter Abbeel, and Sergey Levine. Learning modular neural network policies for multi-task and multi-robot transfer. pp. 2169–2176, 2017. doi: 10.1109/ICRA.2017.7989250. URL https://doi.org/10.1109/ICRA.2017.7989250.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. 2019. URL https://openreview.net/forum?id=SJx63jRqFm.

Haoqiang Fan, Hao Su, and Leonidas J. Guibas. A point set generation network for 3d object reconstruction from a single image. July 2017.

Roy Fox, Sanjay Krishnan, Ion Stoica, and Ken Goldberg. Multi-level discovery of deep options. *arXiv preprint arXiv:1703.08294*, 2017.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35, 2016. URL http://jmlr.org/papers/v17/15-239.html.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. pp. 2672–2680, 2014.

Karol Gregor, George Papamakarios, Frederic Besse, Lars Buesing, and Theophane Weber. Temporal difference variational auto-encoder. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=S1x4ghC9tQ.

Abhishek Gupta, Coline Devin, Yuxuan Liu, Pieter Abbeel, and Sergey Levine. Learning invariant feature spaces to transfer skills with reinforcement learning. 2017. URL https://openreview.net/forum?id=Hyq4yhile.

Donald Hejna, Lerrel Pinto, and Pieter Abbeel. Hierarchically decoupled imitation for morphological transfer. pp. 4159–4171, 2020.

Kuno Kim, Yihong Gu, Jiaming Song, Shengjia Zhao, and Stefano Ermon. Domain adaptive imitation learning. *arXiv preprint arXiv:1910.00105*, 2020.

Taesup Kim, Sungjin Ahn, and Yoshua Bengio. Variational temporal abstraction. In *Advances in Neural Information Processing Systems 32*, pp. 11566–11575. Curran Associates, Inc., 2019. URL http://papers.nips.cc/paper/9332-variational-temporal-abstraction.pdf.

Thomas Kipf, Yujia Li, Hanjun Dai, Vinicius Zambaldi, Alvaro Sanchez-Gonzalez, Edward Grefenstette, Pushmeet Kohli, and Peter Battaglia. Compile: Compositional imitation learning and execution. In *ICML*, 2019.

Sanjay Krishnan, Roy Fox, Ion Stoica, and Ken Goldberg. Ddco: Discovery of deep continuous options for robot learning from demonstrations. *arXiv preprint arXiv:1710.05421*, 2017.

Tejas D. Kulkarni, Karthik R. Narasimhan, Ardavan Saeedi, and Joshua B. Tenenbaum. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pp. 3682–3690, Red Hook, NY, USA, 2016. Curran Associates Inc. ISBN 9781510838819.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. Unsupervised machine translation using monolingual corpora only. 2018a. URL https://openreview.net/forum?id=rkYTTf-AZ.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. Phrase-based & neural unsupervised machine translation. pp. 5039–5049, 2018b. URL https://aclanthology.info/papers/D18-1549/d18-1549.

Ke Li and Jitendra Malik. Implicit maximum likelihood estimation, 2019. URL https://openreview.net/forum?id=rygunsAqYQ.

Fangchen Liu, Zhan Ling, Tongzhou Mu, and Hao Su. State alignment-based imitation learning. 2020. URL https://openreview.net/forum?id=rylrdxHFDr.

Ajay Mandlekar, Yuke Zhu, Animesh Garg, Jonathan Booher, Max Spero, Albert Tung, Julian Gao, John Emmons, Anchit Gupta, Emre Orbay, Silvio Savarese, and Li Fei-Fei. Roboturk: A crowdsourcing platform for robotic skill learning through imitation. In *Conference on Robot Learning*, 2018.

Taesung Park, Alexei A. Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive learning for conditional image synthesis. 2020.

Kaizad V. Raimalwala, B. Francis, and Angela P. Schoellig. A preliminary study of transfer learning between unicycle robots. 2016.

Yannick Schroecker and Charles L Isbell. State aware imitation learning. 30, 2017. URL https://proceedings.neurips.cc/paper/2017/file/08e6bea8e90ba87af3c9554d94db6579-Paper.pdf.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. pp. 86–96, August 2016. doi: 10.18653/v1/P16-1009. URL https://aclanthology.org/P16-1009.

Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, and Sergey Levine. Time-contrastive networks: Self-supervised learning from video. *Proceedings of International Conference in Robotics and Automation (ICRA)*.

Tanmay Shankar and Abhinav Gupta. Learning robot skills with temporal variational inference. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8624–8633. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/shankar20b.html.

Tanmay Shankar, Shubham Tulsiani, Lerrel Pinto, and Abhinav Gupta. Discovering motor programs by recomposing demonstrations. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rkgHY0NYwr.

Archit Sharma, Shixiang Gu, Sergey Levine, Vikash Kumar, and Karol Hausman. Dynamics-aware unsupervised discovery of skills. 2020. URL https://openreview.net/forum?id=HJgLZR4KvH.

Pratyusha Sharma, Lekha Mohan, Lerrel Pinto, and Abhinav Gupta. Multiple interactions made easy (mime): Large scale demonstrations data for imitation. In *CoRL*, 2018.

Xiaolong Wang, Allan Jabri, and Alexei A. Efros. Learning correspondence from the cycle-consistency of time. pp. 2566–2576, 2019. doi: 10.1109/CVPR.2019.00267. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Wang_Learning_Correspondence_From_the_Cycle-Consistency_of_Time_CVPR_2019_paper.html.

Qiang Zhang, Tete Xiao, Alexei A Efros, Lerrel Pinto, and Xiaolong Wang. Learning cross-domain correspondence for control with dynamics cycle-consistency. 2021. URL https://openreview.net/forum?id=QIRlze3I6hX.

Chunting Zhou, Xuezhe Ma, Di Wang, and Graham Neubig. Density matching for bilingual word embedding. In *Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*, Minneapolis, USA, June 2019. URL https://arxiv.org/abs/1904.02343.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

A APPENDIX

You may include other additional sections here.