Neural Policy Translation for Robot Control

Simon Stepputtis[1] Chitta Baral[1] Heni Ben Amor[1]

***Abstract*—Teaching new skills to robots is usually a tedious process that requires expert knowledge and a substantial amount of time, depending on the complexity of the new task. Especially when being used for imitation learning, rapid and intuitive ways of teaching novel tasks are needed. In this work, we outline Neural Policy Translation (NPT) – a novel approach that enables robots to directly learn a new skill by translating natural language and kinesthetic demonstrations into neural network policies.**

Introduction: Humans have the ability to quickly learn a new task based on a single demonstration in combination with an abstract verbal task description. A key component in this process is the ability to generalize from this abstract description to a low-level control policy that generalizes well to variations of the demonstrated task. Natural language is rarely used in current approaches to imitation learning [1], despite being an essential part of human communication. The combination of a physical demonstration, environmental perception and task explanation is the key to efficient learning, which can be expressed in a multimodal vector space. In this work, we present Neural Policy Translation – a methodology that directly creates a control policy from an abstract high-level task description. We use a joint embedding to represent a verbal description, visual environment perception and kinesthetic demonstration of a single task, ultimately allowing us to represent various demonstrations in an efficient joint space. In contrast to the work presented in [2], the goal of our embedding is to capture the necessary information about a variety of different tasks [3] before being used in further inference steps. Subsequently, this embedding is translated into a control policy for the robot by combining it with a similar demonstration of the same task, resulting in the parameters for a sequential low-level control network. By generating these parameters at run-time, the low-level controller is fully flexible and task independent, since the underlying robot control concepts are independent of the high-level task.

Neural Policy Translation: The goal is to learn a general policy $\pi(\mathbf{I}, \mathbf{s}, \mathbf{D})$ that is conditioned on the verbal task description $\mathbf{s} \in \mathbb{R}^5$ and demonstration $\mathbf{D} \in \mathbb{R}^{5\times6}$, resulting in an approximation of each optimal policy π_k^* that would have resulted from directly training the task $\mathcal{T} = \{\tau_1, ..., \tau_k\}$.

1) Multimodal Task Embedding: The task embedding network f_e projects the three separate modality embeddings into a joint space $\mathbb{R}^N$ of length $N = 30$, where each training example $\tau_i \in \mathcal{T}$ is represented as an N-dimensional vector. Before being used in the joint embedding f_e, the variable length trajectory $\mathbf{t} \in \mathbb{R}^{l\times6}$ from each task demonstration τ_i is embedded by converting it into a fixed-length basis representation $\mathbf{D}$. Sentences $\mathbf{s}$ are processed by using a recurrent network that takes separately trained word embeddings $\mathbf{w} \in \mathbb{R}^5$ as input. The visual environment observation $\mathbf{I}$ is converted with a convolutional neural network that is trained as part of the embedding loss $\mathcal{L}_{emb}$ as shown in Equation 1. After having established these distinct representations, the joint embedding f_e is trained based on the difference between the training trajectory $\mathbf{D}$ and *violating trajectory* $\mathbf{D}'$ that was chosen in a pre-processing step to be reasonably different from $\mathbf{D}$ by using Dynamic Time Warping. The overall embedding loss is then formulated as follows, where δ is the mean-squared error and $\triangle$ the cosine similarity:

$$\mathcal{L}_{emb} = \triangle(\mathbf{D}, \mathbf{D}') + \delta\left(f_e\left(\mathbf{s}, \mathbf{I}\right), \mathbf{D}\right) - \delta\left(f_e\left(\mathbf{s}, \mathbf{I}\right), \mathbf{D}'\right) \quad (1)$$

2) Policy Translation: The translation network f_t converts the embedding f_e as well as a trajectory demonstration $\mathbf{D}^*$, that is similar to $\mathbf{D}$, into the parameters Θ_c for the low-level controller f_c. These parameters are then used in the controller to set the *weight* and *bias* values of the neural network [4].

3) Low-Level Control: The low-level controller f_c is a simple sequential model that uses the synthesized parameters Θ_c from the policy translation as weights and bias terms and controls the robot by generating joint velocities.

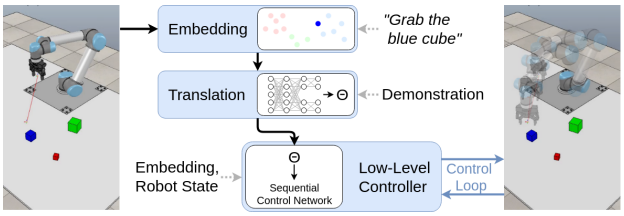

Fig. 1. Schematic overview of the three-staged policy translation network.

Results and Conclusion: The introduced method was evaluated in simulated reaching experiments with two distractors, as shown in figure 1. On unseen testing data, the robot achieves an average joint error of 0.72 degrees with a variance of 0.015 degrees, as compared to the original trajectory. While our method provides promising preliminary results, the ultimate goal of this work is to give robots the ability to understand abstract high-level semantics while being able to perform low-level operations in complex environments.

[1] Y. Duan, M. Andrychowicz, B. Stadie, O. J. Ho, J. Schneider, I. Sutskever, P. Abbeel, and W. Zaremba, "One-Shot Imitation Learning," 2017.

[2] J. Sung, I. Lenz, and A. Saxena, "Deep Multimodal Embedding: Manipulating Novel Objects with Point-clouds, Language and Trajectories," sep 2015.

[3] S. James, M. Bloesch, and A. J. Davison, "Task-Embedded Control Networks for Few-Shot Imitation Learning," oct 2018.

[4] E. Antonios Platanios, M. Sachan, G. Neubig, and T. M. Mitchell, "Contextual Parameter Generation for Universal Neural Machine Translation," tech. rep., 2018.

[1]: Simon Stepputtis, Chitta Baral and Heni Ben Amor are with the School of Computing, Informatics and Decision Systems Engineering, Arizona State University, 660 S. Mill Ave, Tempe, AZ 85281 {sstepput, chitta, hbenamor}@asu.edu