Learning Human Utility from Video Demonstrations for Deductive Planning in Robotics

Nishant Shukla[1], Yunzhong He[1], Frank Chen[1], Song-Chun Zhu[1,2]

(1) Dept. of Computer Science, University of California, Los Angeles, 90024 (2) Dept. of Statistics, University of California, Los Angeles, 90024

Introduction

In this work, we uncouple three components of autonomous behavior (utilitarian value, causal reasoning, and fine motion control) to design an interpretable model of tasks from video demonstrations. The primary contribution of our work include:

1. Learning an *interpretable* utility function independent of the system dynamics
2. *Deductively* exploring goal-reachability under different available actions
3. Proposing *"Fluent Dynamics"* to bridge low-level motion trajectory with high-level utility
4. Teaching a robot how to fold t-shirts, and have it *generalize* to arbitrary articles of clothing

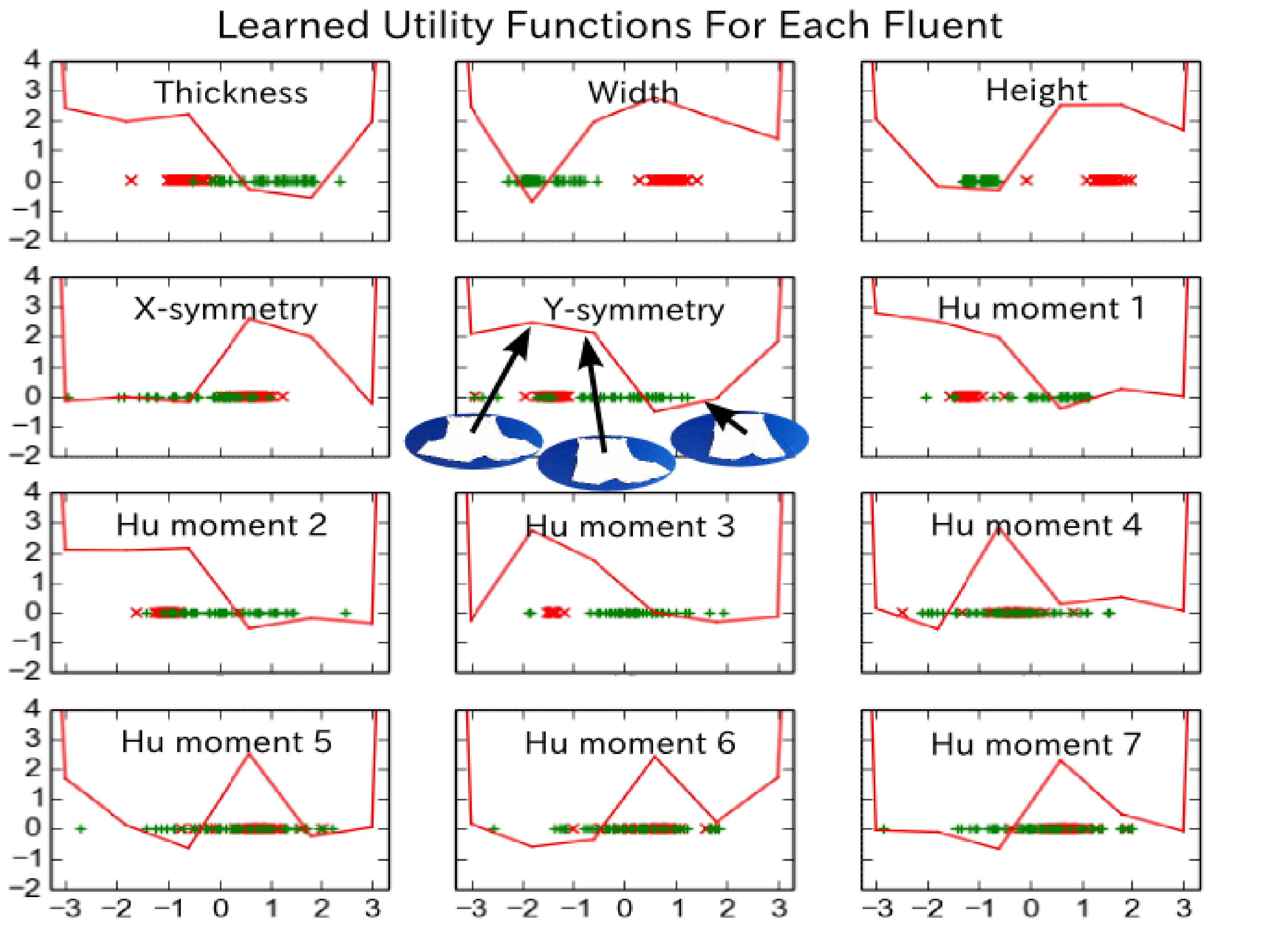

Figure 1: The 12 curves represent the negative utility function ($-\lambda$) corresponding to each fluent. The functions are negated to draw parallels with the concept of potential energy. Red marks indicate fluent values of pg^0, which the learned model appears to avoid, and the green marks indicate fluent values of the goal pg^*, which the learned model appears to favor. Notice how the y-symmetry potential energy decreases as the cloth becomes more and more symmetric. By tracing the change in utilities of each individual fluent, the robot can more clearly explain *why* it favors one state over another.

Model

- An *environment* is defined by generative composition model of objects, actions, and changes in conditions. We use the stochastic context-free And-Or graph (AOG), which models variations and compositions of spatial (S), temporal (T), and causal (C) concepts, called the STC-AOG [1]. The atomic (*terminal*) units of this composition grammar are tuples of the form ($F_{start}, u_{[1:t]}, F_{end}$), where F_{start} and F_{end} are pre- and post-fluents of a sequence of interactions $u_{[1:t]}$.
- A *state* is a configuration of the believed model of the world. We represent state as a parse graph (pg) of the And-Or graph. The set of all parse-graphs is denoted Ω_{pg}.
- A *fluent* is a condition of a state that can change over time [Mueller, 2014]. We represent it as a real-valued function on the state: $f_i : \Omega_{pg} \to \mathbb{R}$.
- A *fluent-vector* F is a column-vector of fluents: $F = (f_1, f_2, \dots, f_k)^T$.
- A *goal* is characterized by a fluent-change ΔF. The purpose of learning the utility function is to identify reasonable goals.

The Utility Landscape

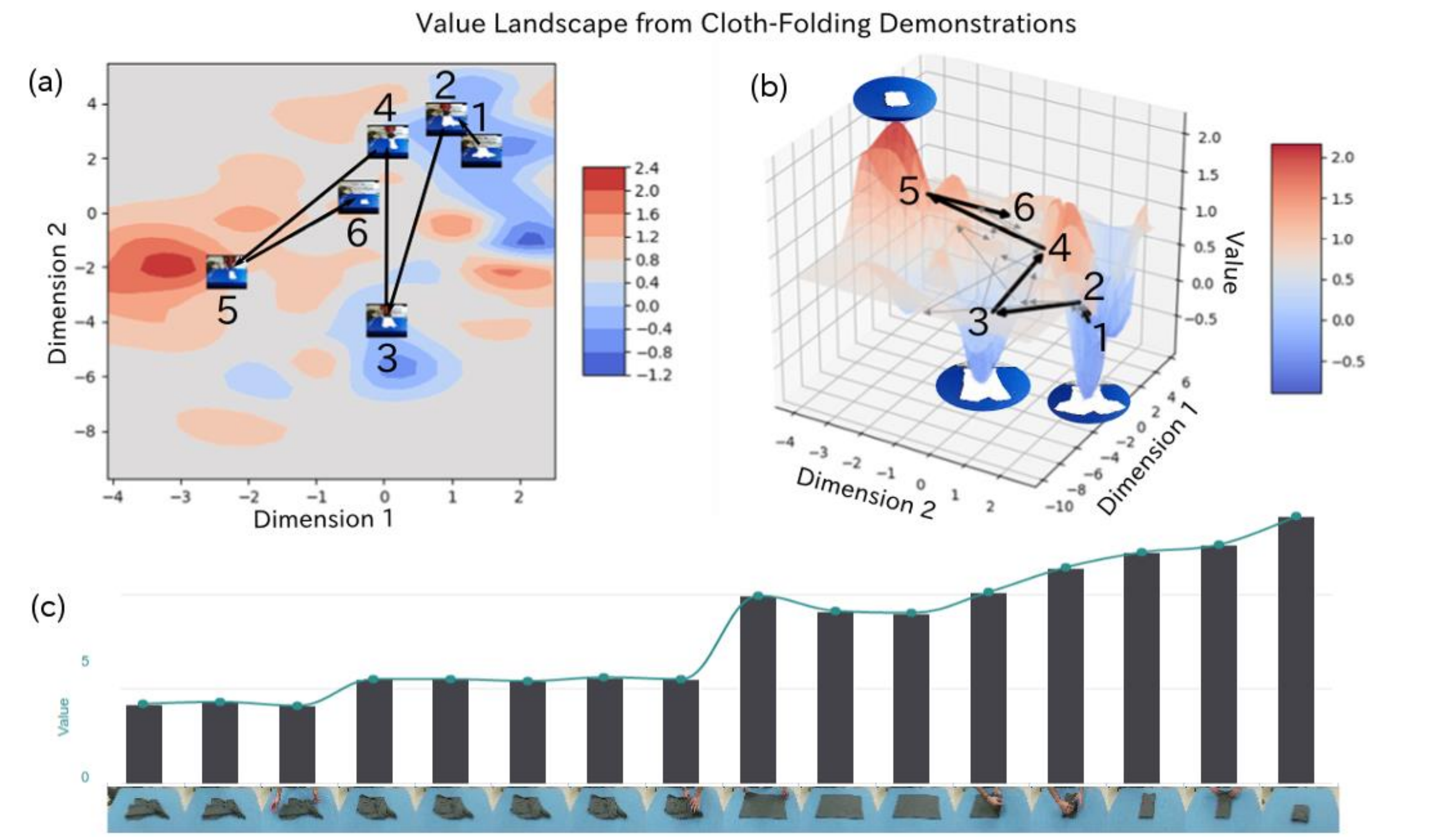

Figure 2: These landscapes are trained from 45 cloth-folding video demonstrations. (a) shows the 2D landscape; (b) shows the canyons for wrinkled clothes, and the peaks for well-folded clothes; (c) shows the gradual increase in utility of the shirt.

Learning Utility

- We assume human preferences are derived from a utilitarian model, in which a latent utility function assigns a real-number to each configuration of the world. We use a maximum margin formulation to select relevant fluents by minimizing the ranking violations of the model.
- If a state pg^1 has a higher utility than another state pg^2, then the ranking is denoted $pg^1 \succ pg^2$, implying that pg^1's utility is greater.
- Denote $\Lambda = \{\lambda^{(1)}(), \lambda^{(2)}(), \dots, \lambda^{(K)}()\}$ as the corresponding set of utility functions for each fluent in F.
- The total utility function is:

$$U(pg; \Lambda, F) = \sum_{\alpha=1}^{K} \lambda^{(\alpha)}(f^{(\alpha)}(pg))$$

- The optimization problem for ranking violations becomes:

$$\min \sum^{K} \int_{x} \lambda''^{(\alpha)} dx + C \sum_{v} \xi_v$$

$$s.t. \sum_{\alpha} \left(\lambda^{(\alpha)} \left(f^{(\alpha)}(pg_v^*) \right) - \lambda^{(\alpha)} \left(f^{(\alpha)}(pg_v^0) \right) \right) > 1 - \xi_v,\ \xi_v \geq 0.$$

Fluent Dynamics

- We define the cost of a sequence of actions $V(a^*_{[1:t]})$ by the utility of the resulting fluent vector.
- The robot must use its available actionable information to maximize utility.

$$a^*_{[1:t]} = \arg\max_{a_{[1:t]}} V(a_{[1:t]})$$

- Optimal action sequences will satisfy $\frac{\partial V}{\partial a_{[1:t]}} = 0$. The gradient of V with respect to $a_{[a:t]}$ can be computed using the chain rule.

$$\frac{\partial V}{\partial a_{[1:t]}} = \frac{\partial V}{\partial F} \frac{\partial F}{\partial u_{[1:t]}} \frac{\partial u_{[1:t]}}{\partial a_{[1:t]}}$$

- $\frac{\partial V}{\partial F}$ comes from utility learning
- $\frac{\partial F}{\partial u_{[1:t]}}$ is solved using the assumed And-Or compositional model of the world
- $\frac{\partial u_{[1:t]}}{\partial a_{[1:t]}}$ is solved using inverse kinematics and optimal control methods [2]

Implementation & Results

- We learned our utility function from 45 RGB-D (pointcloud) video demonstrations of t-shirt folding. This dataset splits into 30 for training and 15 for testing.
- At each frame, the vision processing step segments the cloth using graph-based techniques [3,4] and align the 3D cloth pointcloud to its principal axis
- Next, we extract fluents from the pointcloud being tracks.
- Examples of fluents include width, height, thickness, x-symmetry, y-symmetry, and 7 moment invariants [5]
- We cross validate the fitting of the learned utility model
- We also perform external evaluations on 330 individuals to compare how well the learned preference model matches human judgement.

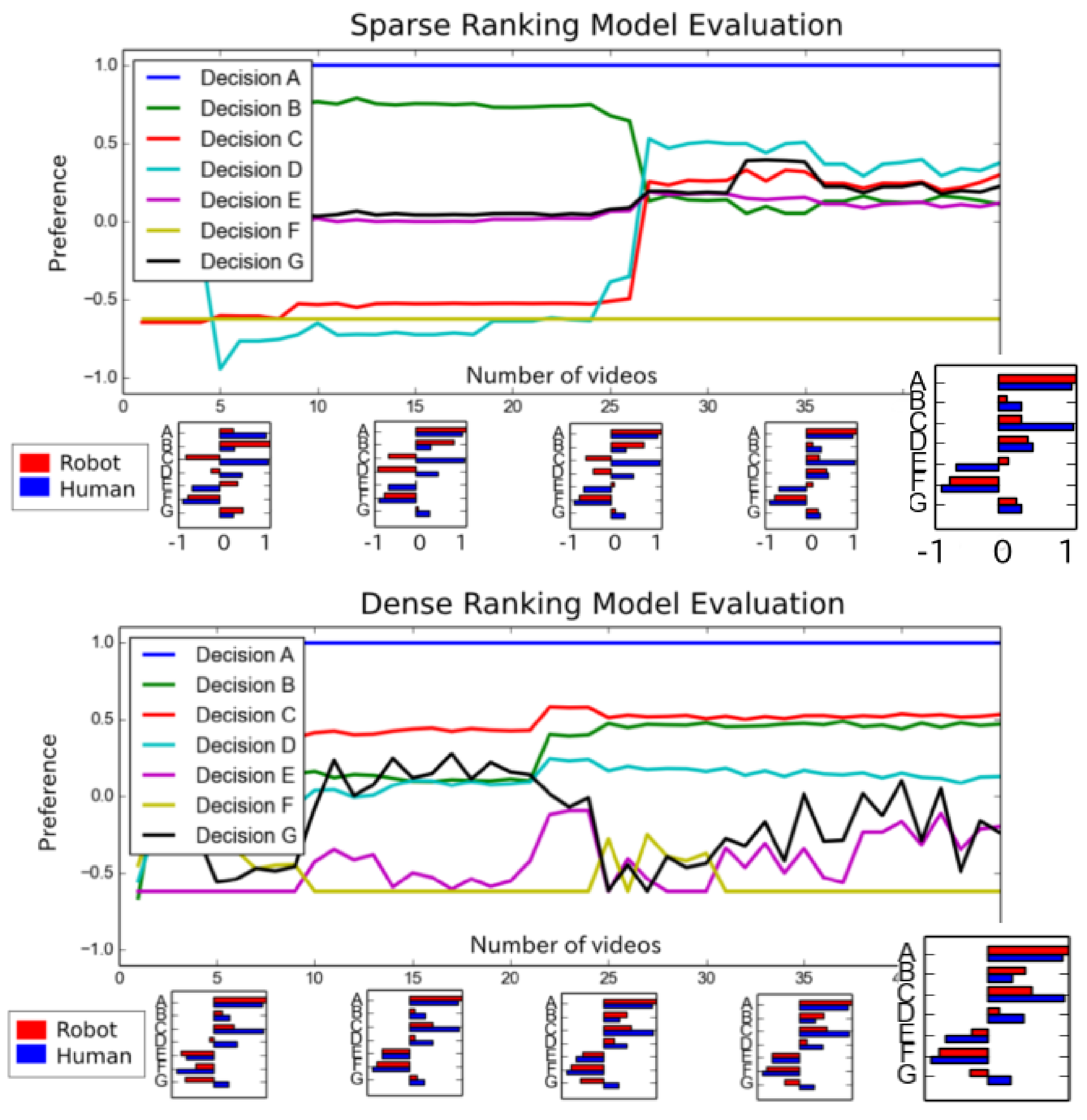

Figure 3: The sparse and dense ranking models are evaluated by how quickly they converge and how strongly they match human preferences. The x-axis on each plot indicates the number of unique videos shown to the learning algorithm. The y-axis indicates two alternatives (1 vs. -1) for 7 decisions (A, B, C, D, E, F, and G) of varying difficulty. The horizontal bar-charts below each plot show comparisons between human and robot preferences. As more videos are made available, both models improve performance in convergence as well as alignment to human preferences (from 330 survey results).

Acknowledgements

This work is supported by DARPA-XAI Project Award No, N66001-17-2-4029. We would like to explicitly thank Yixin Zhu, Mark Edmonds, Hangxin Liu, and Feng Gao for their help with the robot platform.

Citations:

[1] C. Xiong, N. Shukla, W. Xiong, and S. Zhu. Robot learning with spatial, temporal, and causal and-or graph. In *Proceedings of International Conference on Robotics and Automation (ICRA 2016)*, Stockholm, Sweden, May 2016.
[2] Y. Tassa, N. Mansard, and E. Todorov. Control-limited differential dynamic programming. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 2224-2231, 2013.
[3] P.F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *Int. J. Comput. Vision*, 59(2):167-181, Sept. 2004. ISSN 0920-5691.\
[4] C. Rother, V. Kolmogorov, and A. Blake. "grabcut": Interactive foreground extraction using iterated graph cuts. ACM Trans. Graph. 23(3):309-314, Aug. 2004. ISSN 0730-0301.
[5] M.-K. Hu. Visual pattern recognition by moment invariants. *IRE transactions on information theory*, 8(2):179-187, 1962.