A Dynamic Neural Network Approach to Generating Robot's Novel Actions: A Simulation Experiment

[1]Korea Advanced Institute of Science and Technology, Korea
[2]Okinawa Institute of Science and Technology, Japan
Jungsik Hwang[1,2] and Jun Tani[2]

Research Objectives

Generating Robot's Novel Actions
- From experience of learning basic actions

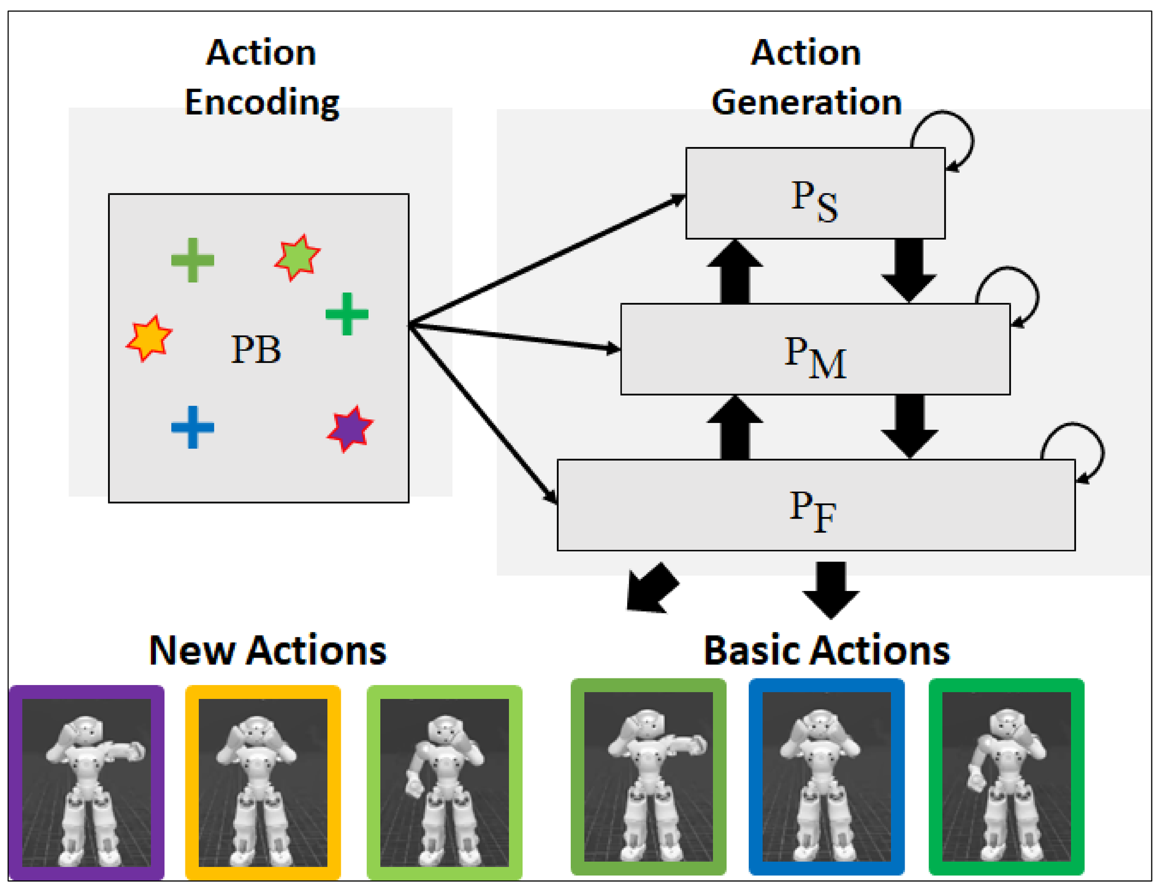
Dynamic Neural Network Approach
- Encoding actions into its own memory
- Non-linear Memory → Source of Novelty

Key Findings

Generation of Novel/Creative Actions
- By modulating & combining the learned actions
- Emerged from non-linear memory structure

Learning Method Influenced the Level of Creativity
- By inducing self-organization of memory structure with different characteristics

Multiple Timescales RNN with Parametric Biases



Network Architecture

Network Architecture
- **[Action Generation Module]** MTRNN for learning/generating robot's sequential behavior
- **[Action Encoding Module]** PB for mapping robot's high-dimensional action to low-dimensional space
- **One pair of PB values represents a single action**

Key Features
- Encoding actions in the continuous PB space without human intervention
- Generating robot's action without any external information, but only with given PB values (i.e. mental simulation)

Learning/Generating Actions

Learning Actions during Training
- **[Dataset]** Obtained from tutoring (LfD)
- **[Supervised Training]** Trained to generate 1-step prediction of joint angle values
- Optimize Weights/Biases and **PB Values** ($\frac{\partial E}{\partial PB}$)
- ***N pairs of PB values for N data**

Open/Closed-Loop Learning
INPUT(t) = γ·OUTPUT(t-1) + (1- γ)·DATASET(t)
- Open-Loop Training: γ = 0.0
- Closed-Loop Training: γ = 1.0

Generating Actions during Testing
- Closed-loop generation with given PB values
- No external input is required

Results

Experiment Settings
- **Robotic Platform**
  - NAO (Simulation): 4 x 2 DoFs
  - 6 Boxing-like actions
- **Network Configuration** ($PB/P_S/P_M/P_F$)
  - # of neurons : 2 /10 / 20 / 40
  - Time constants: - / 8 / 4 / 2

Training the Model
- 3 Training Conditions
  - Open-Loop / Closed-Loop / Half Closed-Loop (γ = 0.5)
- ADAM / Tensorflow / 100,000 epochs

Generating Actions
- By linearly sampling 200 x 200 PB values
  - PB = Linspace(-1, 1, 200)

Measuring the Level of Creativity
- **Appropriateness**
  - Neither too fast nor too slow
- **Novelty**
  - *Dist* (Generated Actions , Learned Actions)
- **Diversity**
  - *Dist* (Generated Actions, Generated Actions)

Level of Creativity in terms of 3 Measures

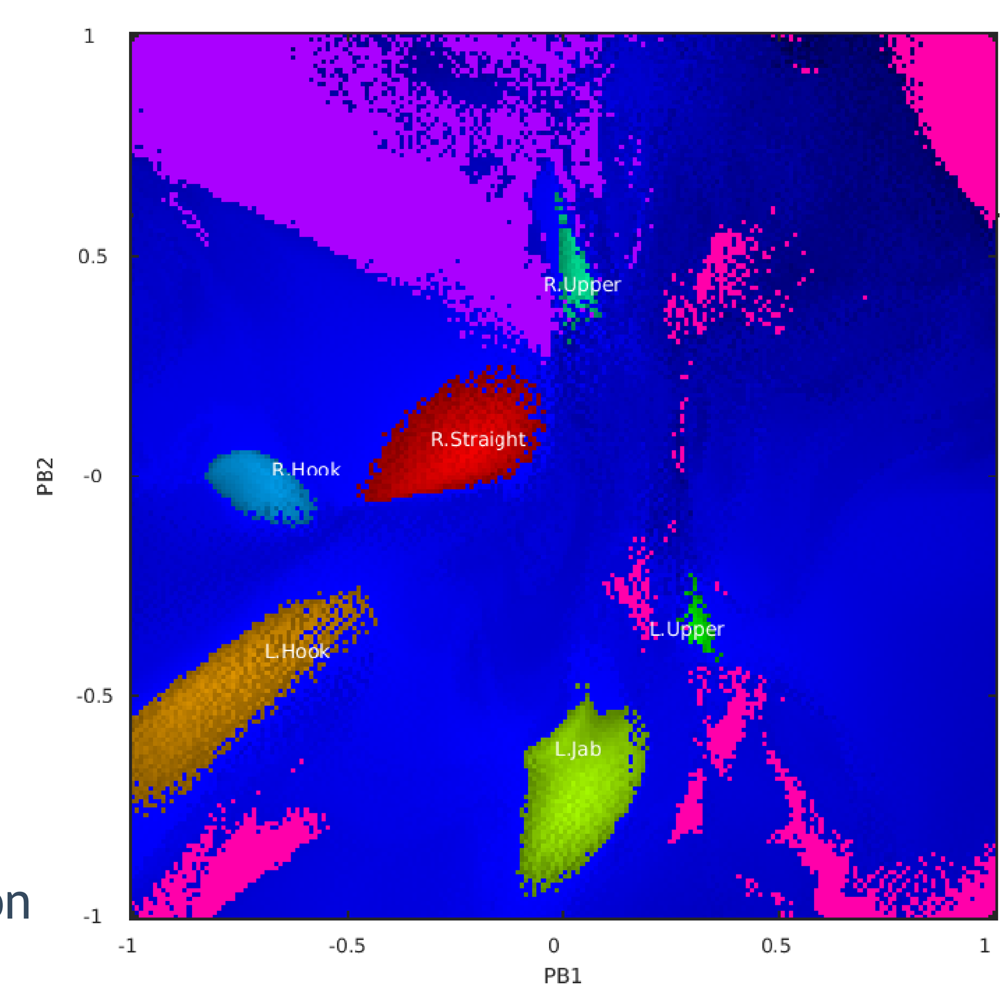
		Closed-loop Ratio (γ) during Training		
		0.0	0.5	1.0
Appropriateness (%)	Unlearned	72.21	75.26	57.95
	Learned	11.23	7.58	40.02
	Subtotal	83.44	82.84	97.97
Novelty		26.02	31.71	18.53
Diversity		43.12	48.03	35.96

Non-Linear Memory Structure Self-Organized at the Action Encoding Module

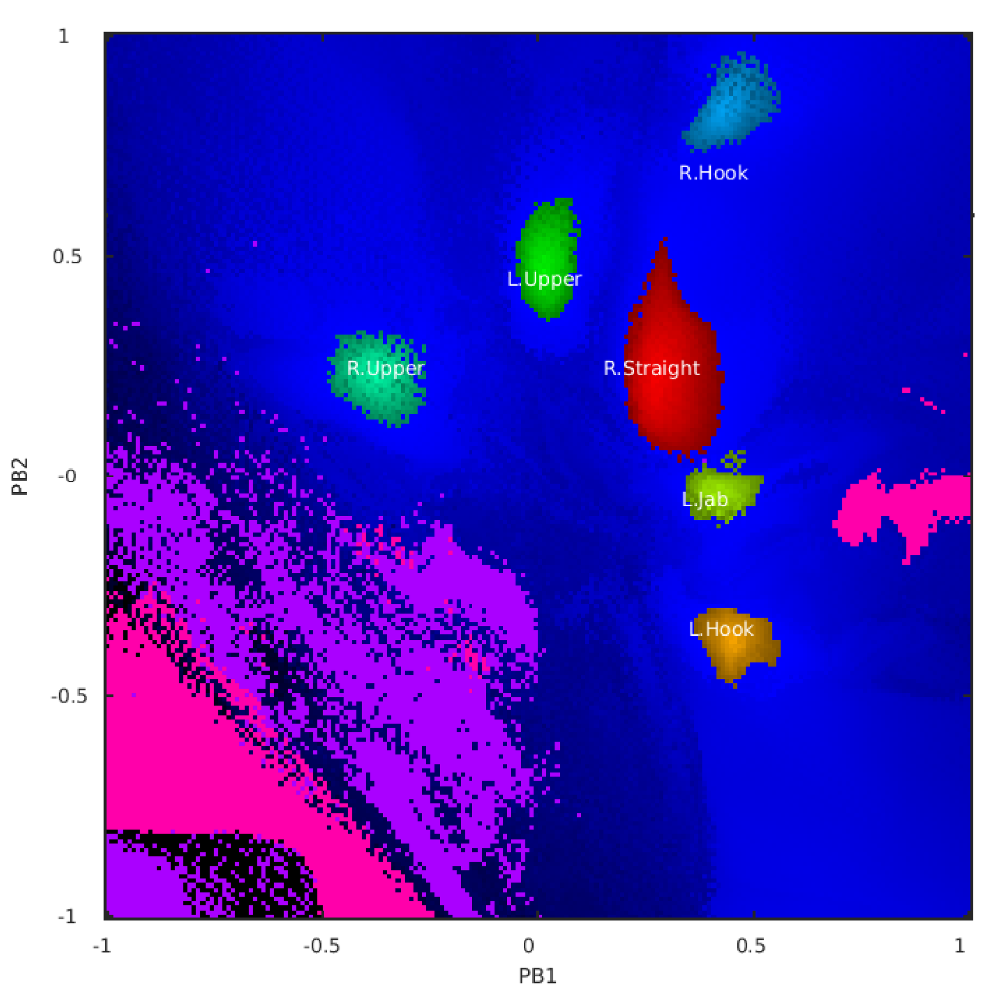
Visualization
- 200 x 200 PB values
- Each value encodes single action

Color Code
- 6 Learned actions
- Novel Actions
- Too Fast
- Not Moving

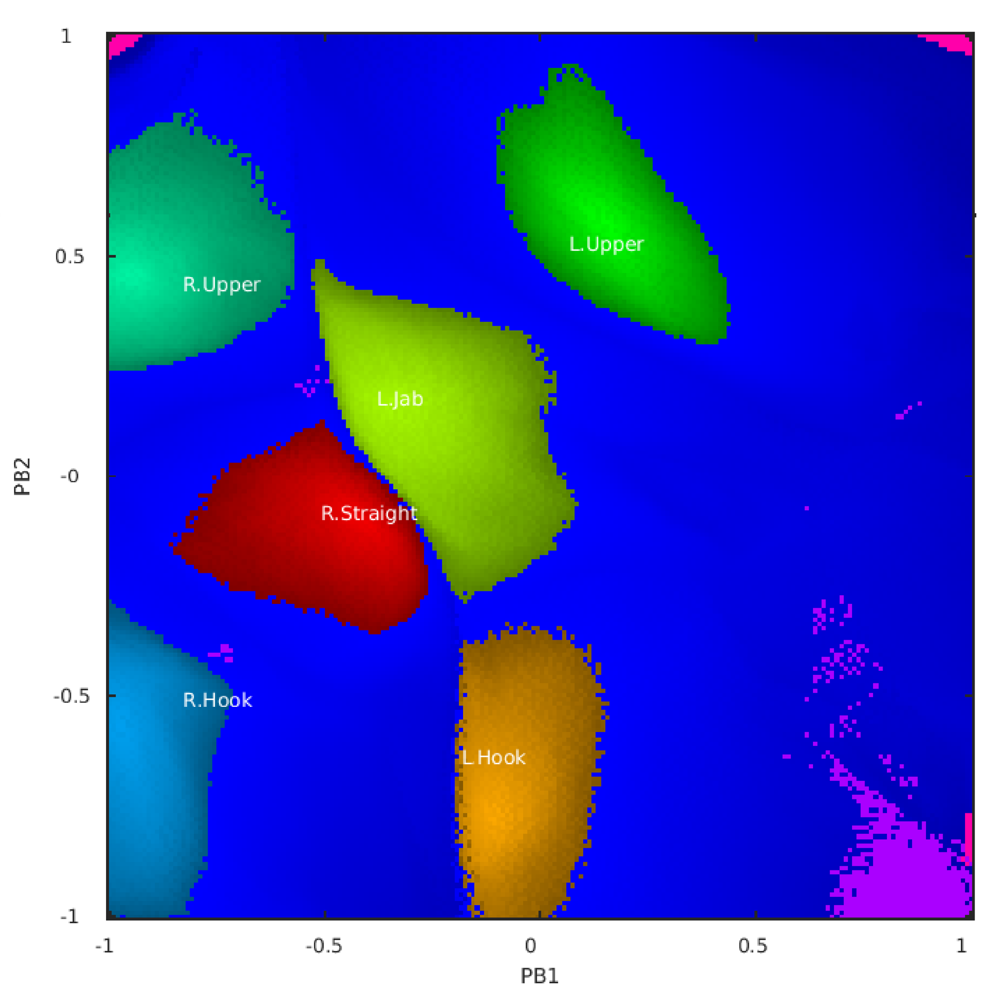
Results
- "Rugged" Landscape
- Small changes in PB
  → Huge changes in Action
- **Source of Novelty**



Open-Loop Training



Half Closed-Loop Training



Closed-Loop Training