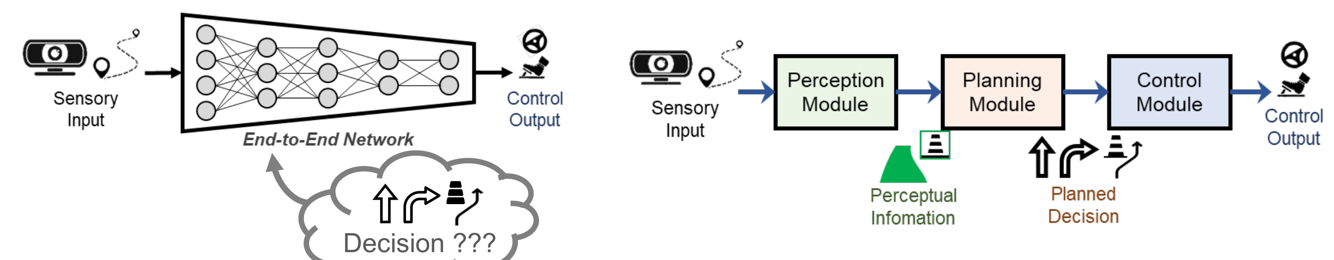
Self-Supervised Interpretable End-to-End Learning via Latent Functional Modularity

Hyunki Seong, David Hyunchul Shim
Korea Advanced Institute of Science and Technology

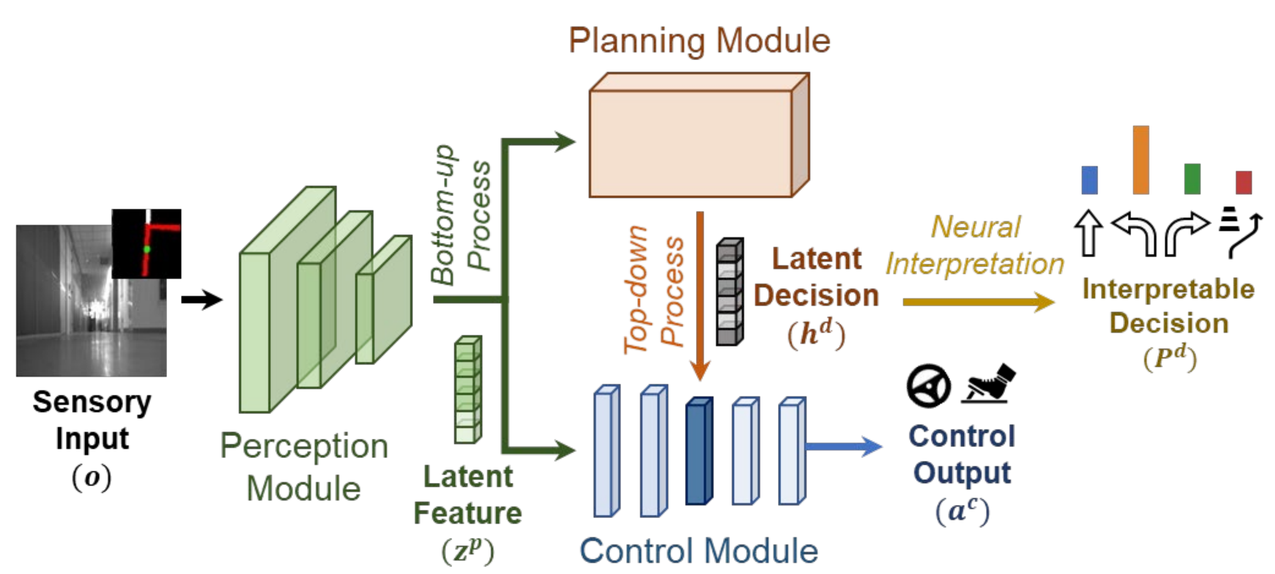
Motivation



Recent End-to-End Architectures:
- ✓ Efficient to learn policy
- ✓ No (or Less) heuristic
- ✓ Struggle to learn task-specific policy
- ✓ No interpretable process

Rethinking Modular Architectures:
- ✓ Task-oriented control
- ✓ Interpretable representation
- ✓ Complex inter-module dependencies
- ✓ Require much heuristics

Contributions



Main Contributions:

1) ***End-to-End Architecture** + **Latent Functional Modularity***
2) ***Self-Supervised Sensorimotor Learning** with **Task Specificity***
3) ***Neural Interpretation** via a **Post-Hoc Explainability Method***

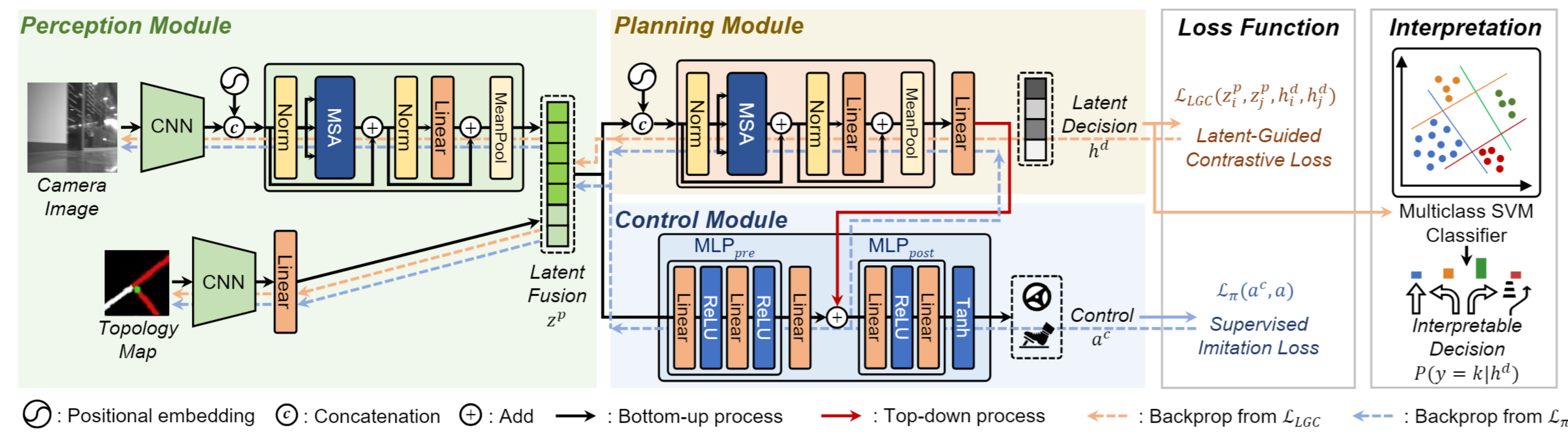
Discussion

Our interpretable end-to-end learning
- ✓ Has **more *reliable*** and **less *uncertain*** sensorimotor control
- ✓ Facilitates a ***hybrid architecture*** (end-to-end + external modules)
- ✓ Integrages ***robotic learning*** with ***eXplainable Artificial Intelligence***

Paper & Code

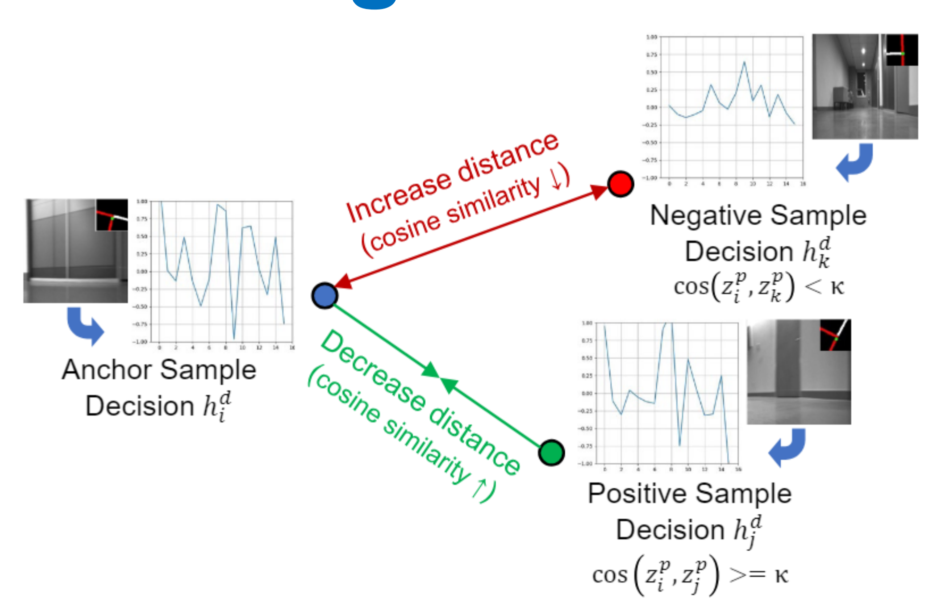
USRG Lab

Overview



1) **Network Architecture:** Modular End-to-End Network (***Perception*** & ***Planning*** & ***Control***)
2) **Training Objectives:** Latent-Guided Contrastive Loss ($\mathcal{L}_{LGC}$) + Supervised Imitation Loss ($\mathcal{L}_\pi$)
3) **Neural Interpretation Methods:** Post-hoc Multiclass SVM Classifier + Calibration Method

Training Details



$$\mathcal{L}_{\text{LGC}} = \begin{cases} 1 - \cos(h_i^d, h_j^d) & \text{if } \cos(z_i^p, z_j^p) >= \kappa \\ \max(0, \cos(h_i^d, h_j^d)) & \text{if } \cos(z_i^p, z_j^p) < \kappa \end{cases}$$

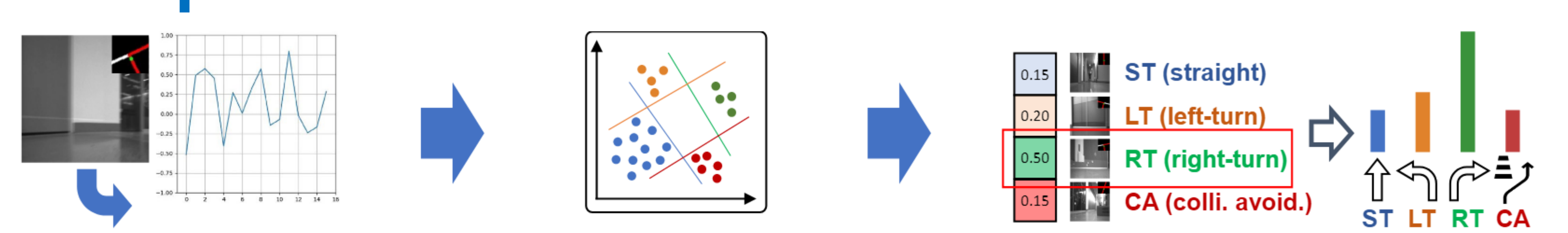
$$\mathcal{L}_\pi(a^c, a) = |\delta^c - \delta| + \lambda_\tau |\tau^c - \tau|$$

$$\mathcal{L} = \mathcal{L}_\pi(a_i^c, a_i) + \lambda_{\text{LGC}} \mathcal{L}_{\text{LGC}}(z_i^p, z_j^p, h_i^d, h_j^d)$$

Encourage the ***Planning*** module to generate similar latent decisions from similar perceptual features z^p (*positive samples*), while producing diverse decisions from dissimilar driving contexts z^p (*negative samples*).

→ ***Self-Supervised Task-Specific Sensorimotor Learning without Task-Level Supervision***

Interpretation Details



1. Generate a latent decision
2. Classify the task of the latent decision using a trained Multiclass SVM Classifier
3. Calibrate the probabilities of the SVM Classifier's results to yield more interpretable representations $P(y=k|h^d)$

$$\min_{w,b} \frac{1}{2} w^T w + C \sum_{i=1}^{M} max(0, 1 - y_i(w^T h_i^d + b))^2$$

▲ Optimization problem for Support Vector Machine

$$P(y_i = k | h_i^d) = \frac{1}{1 + exp(E_k f_k(h_i^d) + F_k)}$$

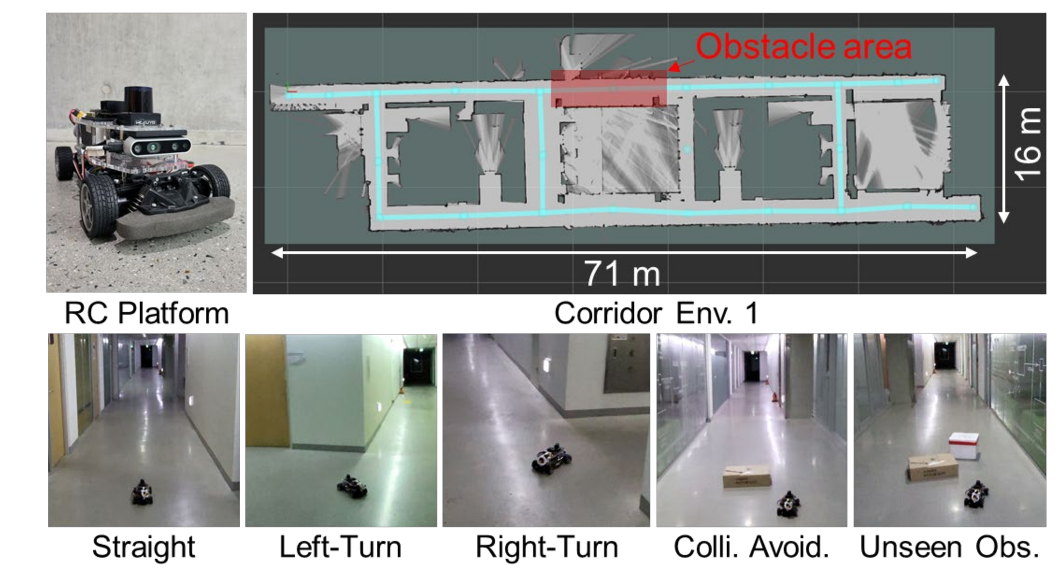
▲ Calibration method using sigmoid function

- ✓ The **latent decision** vector is decoded through **lightweight Multiclass Support Vector Machines (SVMs)**
- ✓ Utilize a **calibration method** to represent the classification results as ***probabilities among different tasks***

→ ***Mapping the Mind (Decision) of a Robotic End-to-End Sensorimotor Network***

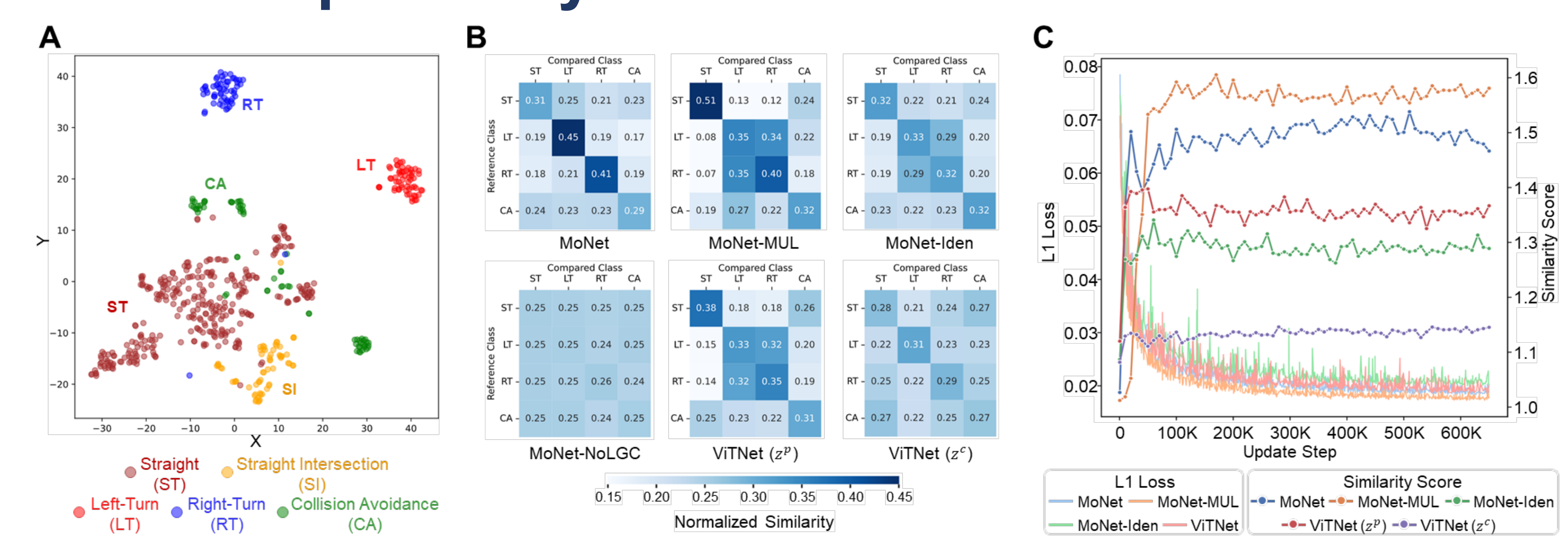
Experiments

❑ Hardware & Scenario Setup



- ✓ A robotic RC vehicle platform
- ✓ Indoor navigation scenarios
- ✓ Multiple driving tasks
  - Straight (**ST**)
  - Left-Turn (**LT**)
  - Right-Turn (**RT**)
  - Collision Avoidance (**CA**)

❑ Task Specificity Results



t-**SNE** Visualization | **R**epresentational **S**imilarity **M**atrix | Learning Curves

❑ Neural Interpretation Results