Adaptive RBF Networks for Robust Feature Selection in Inverse Reinforcement Learning

Ashiq CS22B2021
Ajmal CS22B2046
Abishek Chakravarthy CS22B2054

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, DESIGN AND MANUFACTURING, KANCHEEPURAM

Introduction to IRL

Problem Statement:

- Learn a reward function from expert demonstrations.
- Use Radial Basis Function (RBF) networks to model rewards.

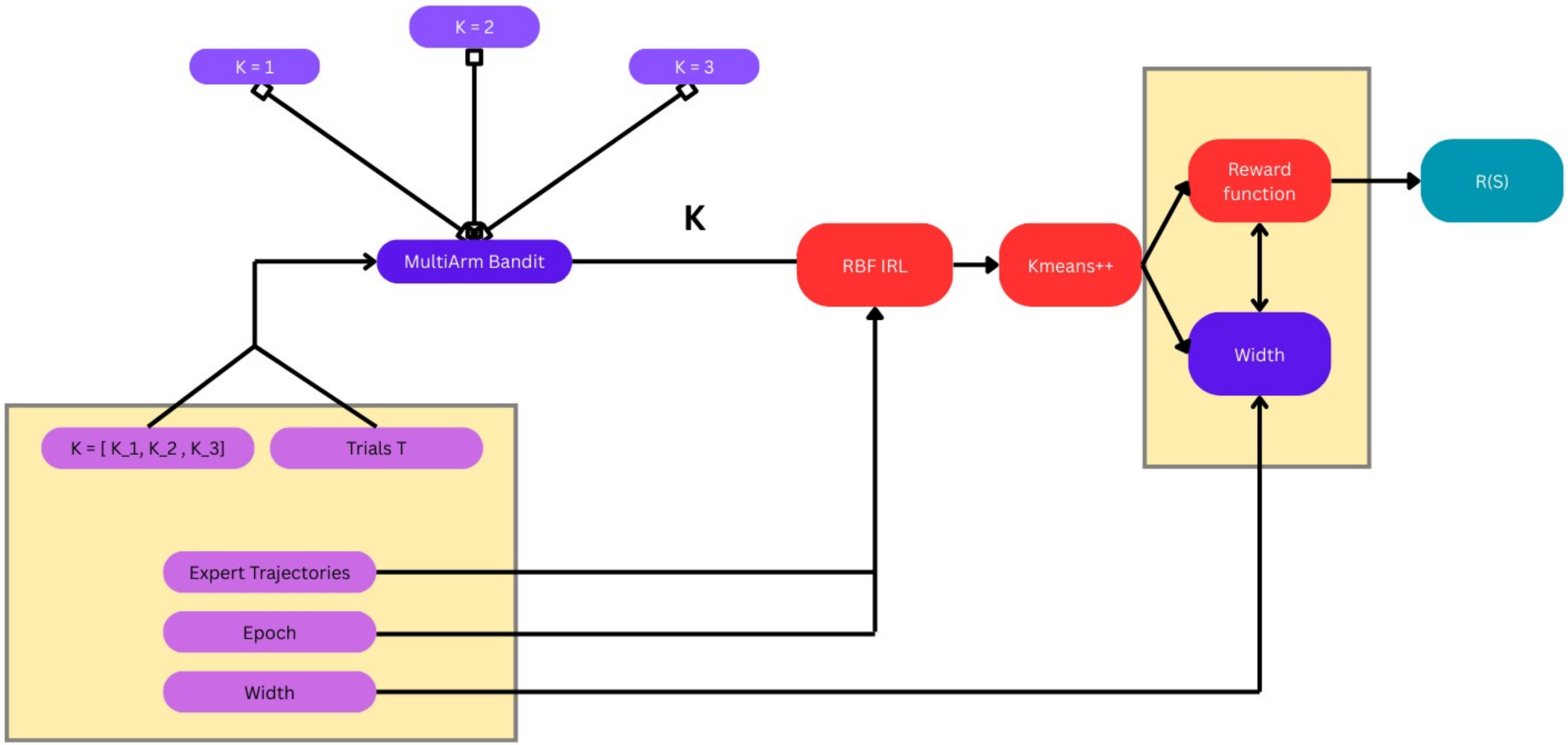
Key Challenges:

- Optimal selection of RBF centers (K).
- Adaptive kernel width tuning.

Methodology Overview

1) **Multi-Armed Bandit (MAB):**
    1) UCB algorithm to select optimal K for RBF centers
    2) Balances exploration vs. exploitation.
2) **RBF Network Components:**
    1) Centers (K-means clustering)
    2) Kernel widths (adaptive, per-cluster, learned)
3) **Training:**
    1) Match feature expectations between expert and learned policies

Architecture Overview



Multi-Armed Bandit for K Selection

1) BanditKSelector Class:

1) Input: Candidate K values (e.g., [5, 10, 15])
2) Reward: Silhouette score (cluster quality).

2) UCB Formula (exploitation vs exploration)

1) UCB = exploitation + exploration_weight * sqrt(log(total_counts)/counts[k])

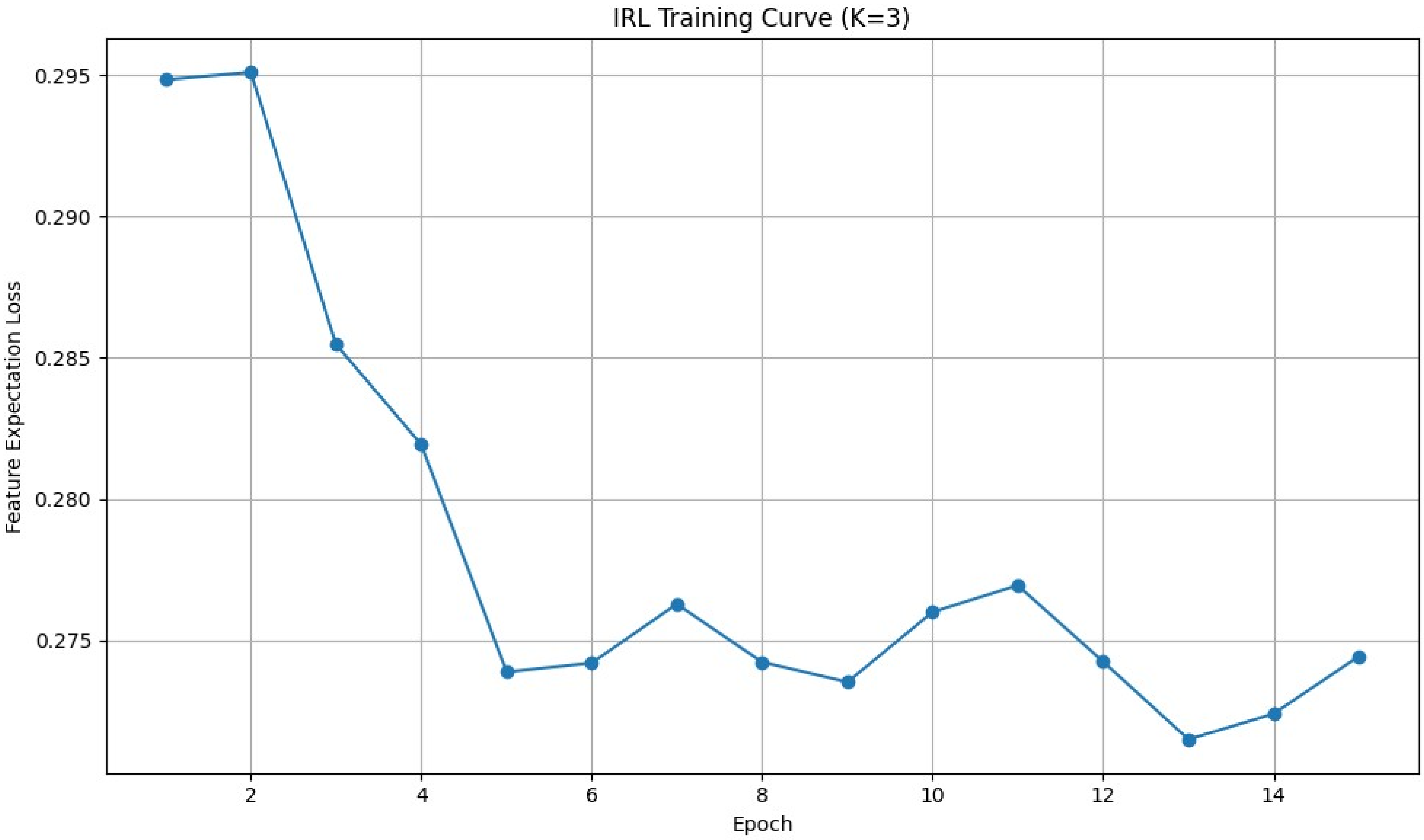
Hyperparameter selection for **k** in k-means clustering.

Selection for K

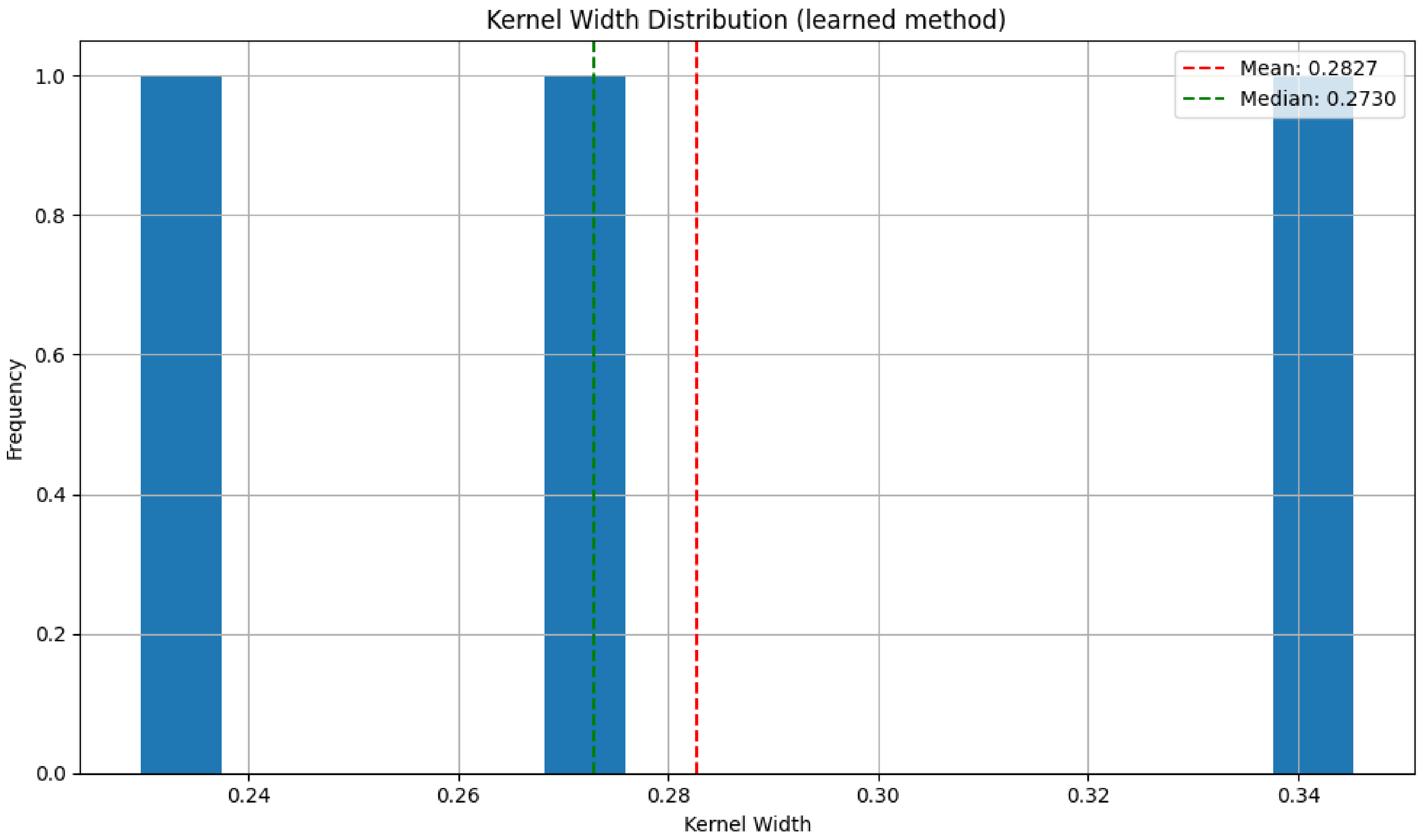
```
Training with Learned Kernel Widths...
Selecting K using multi-armed bandit and expert data clustering quality...
Trial 1/20 for K=2, Silhouette Score: 0.8178
Trial 2/20 for K=3, Silhouette Score: 0.8534
Trial 3/20 for K=4, Silhouette Score: 0.8194
Trial 4/20 for K=5, Silhouette Score: 0.8214
Trial 5/20 for K=3, Silhouette Score: 0.8403
Trial 6/20 for K=5, Silhouette Score: 0.8236
Trial 7/20 for K=4, Silhouette Score: 0.8116
Trial 8/20 for K=2, Silhouette Score: 0.8021
Trial 9/20 for K=3, Silhouette Score: 0.8517
Trial 10/20 for K=5, Silhouette Score: 0.8139
Trial 11/20 for K=4, Silhouette Score: 0.8161
Trial 12/20 for K=2, Silhouette Score: 0.8160
Trial 13/20 for K=3, Silhouette Score: 0.8443
Trial 14/20 for K=5, Silhouette Score: 0.8213
Trial 15/20 for K=4, Silhouette Score: 0.8185
Trial 16/20 for K=2, Silhouette Score: 0.8205
Trial 17/20 for K=3, Silhouette Score: 0.8419
Trial 18/20 for K=5, Silhouette Score: 0.8184
Trial 19/20 for K=4, Silhouette Score: 0.8155
Trial 20/20 for K=2, Silhouette Score: 0.8132
```

Given possible k = 2,3,4,5

For K = 3


IRL Training Curve (K=3)
Feature Expectation Loss
0.295
0.290
0.285
0.280
0.275
2
4
6
8
10
12
14
Epoch

For K = 3



Kernel Width Distribution (learned method)
Mean: 0.2827
Median: 0.2730
Frequency
0.0
0.2
0.4
0.6
0.8
1.0
0.24
0.26
0.28
0.30
0.32
0.34
Kernel Width

RBF Kernel Width Methods

1) Adaptive:

   Widths based on cluster density and nearest-center distance.

2) Per-Cluster:

   Median distance of points to cluster center.

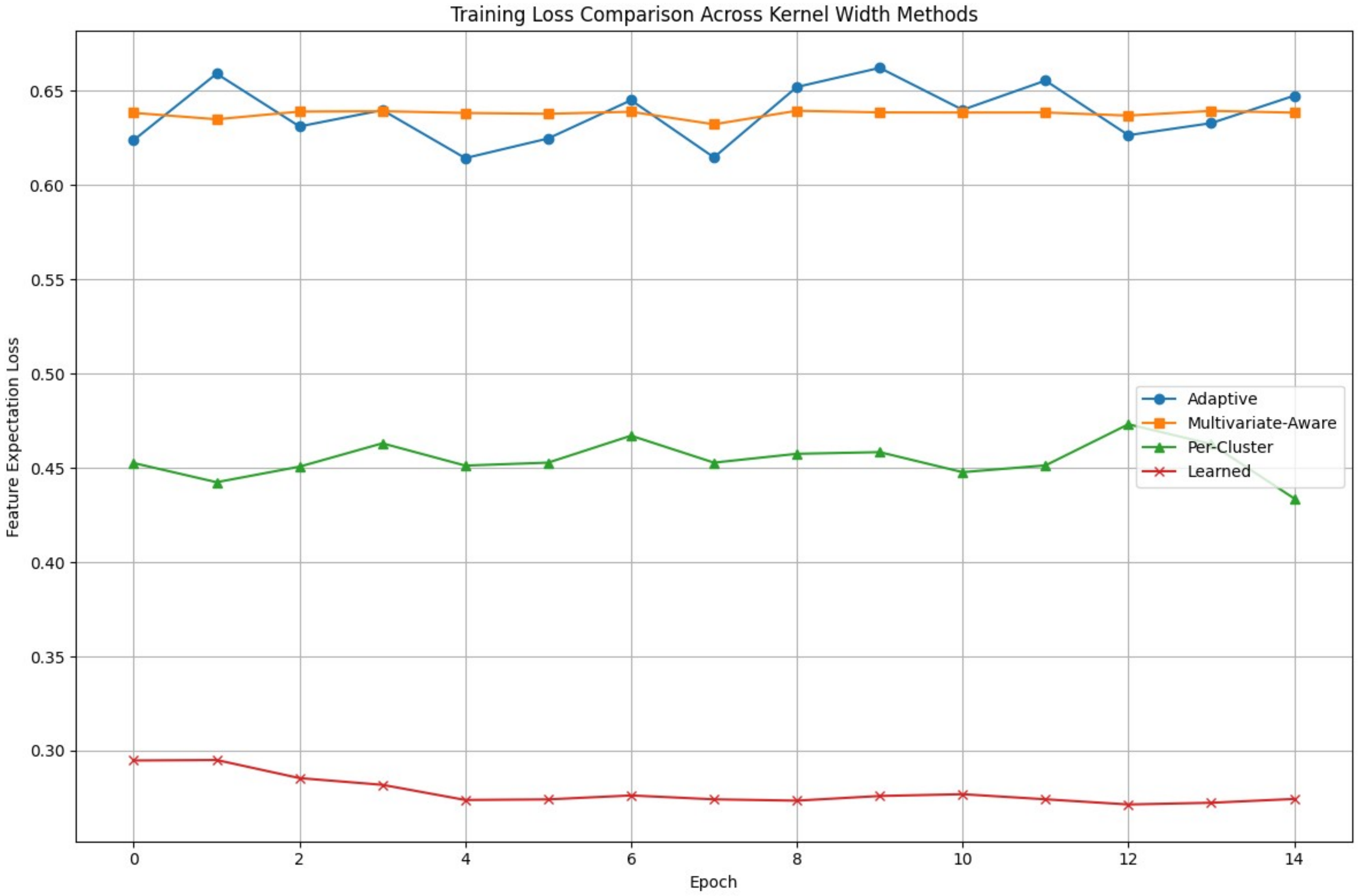
3) Multivariate-Aware:

   Covariance matrix for high-dimensional state spaces.

4) Learned:

   Optimized during training via reward difference.

Comparison Graph



Training Process

Algorithm:

1) Center Determination:
    1) If using K-selection, run bandit algorithm to pick optimal K
    2) Run K-means to determine RBF centers based on expert states
    3) Compute kernel widths using the selected method

Training Process 2

Iterative Policy Improvement:

1) Initialize weights randomly
2) For each epoch:
    1) a. Collect rollouts using current reward function
    2) b. Compute feature expectations for both expert and rollout trajectories
    3) c. Update weights using gradient: w += learning_rate * (expert_features - rollout_features)
    4) d. If using learned widths, adjust them to maximize the reward difference between expert and non-expert states
    5) e. Compute and track feature expectation loss

Training Process 3

Reward Optimization:

1) The objective is to minimize the difference between expert and policy feature expectations
2) L2 regularization can be applied to prevent overfitting

Policy Learning

After learning the reward function, a reinforcement learning algorithm (PPO in this case) is used to learn a policy that maximizes the learned reward.

Training Process 4

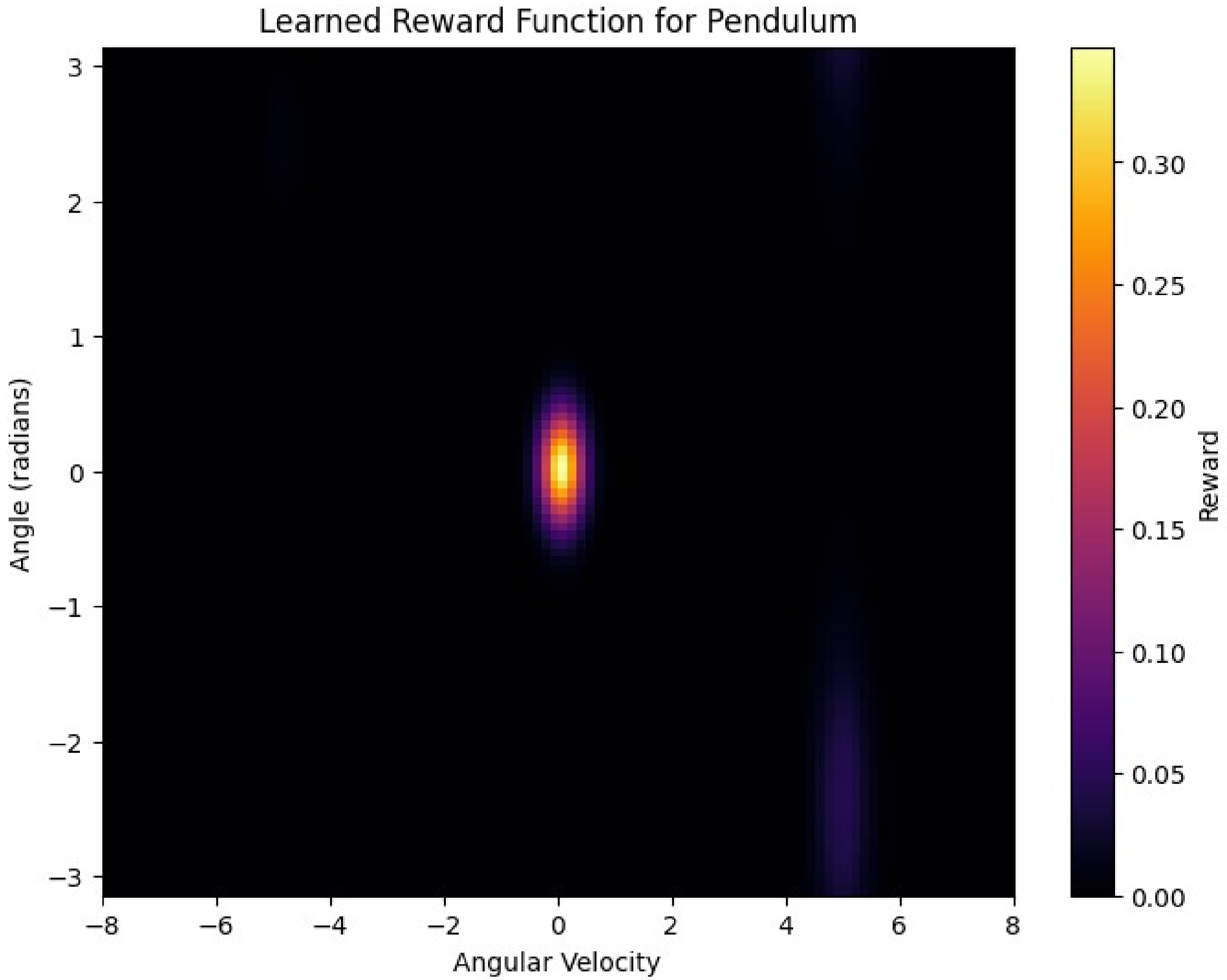
```
Selected K=3 (best silhouette score: 0.8534)
Computed 3 RBF centers with learned kernel widths
Kernel widths range: 0.0653 to 1.3721
Mean width: 0.8511, Std dev: 0.5654

Starting IRL training...
Epoch 1/15, Feature Expectation Loss: 0.2948
Epoch 2/15, Feature Expectation Loss: 0.2951
Epoch 3/15, Feature Expectation Loss: 0.2855
Epoch 4/15, Feature Expectation Loss: 0.2819
Epoch 5/15, Feature Expectation Loss: 0.2739
  Kernel width range: 0.1052 to 0.8102
Epoch 6/15, Feature Expectation Loss: 0.2742
Epoch 7/15, Feature Expectation Loss: 0.2763
Epoch 8/15, Feature Expectation Loss: 0.2742
Epoch 9/15, Feature Expectation Loss: 0.2735
Epoch 10/15, Feature Expectation Loss: 0.2760
  Kernel width range: 0.1695 to 0.4784
Epoch 11/15, Feature Expectation Loss: 0.2769
Epoch 12/15, Feature Expectation Loss: 0.2742
Epoch 13/15, Feature Expectation Loss: 0.2715
Epoch 14/15, Feature Expectation Loss: 0.2724
Epoch 15/15, Feature Expectation Loss: 0.2744
  Kernel width range: 0.2298 to 0.3453
```

Experiments & Results

Pendulum

- Compared Methods:
  - Adaptive, Per-Cluster, Multivariate, Learned.
- Key Results:
  - Learned widths achieve lowest training loss.
  - Adaptive methods balance speed and performance.

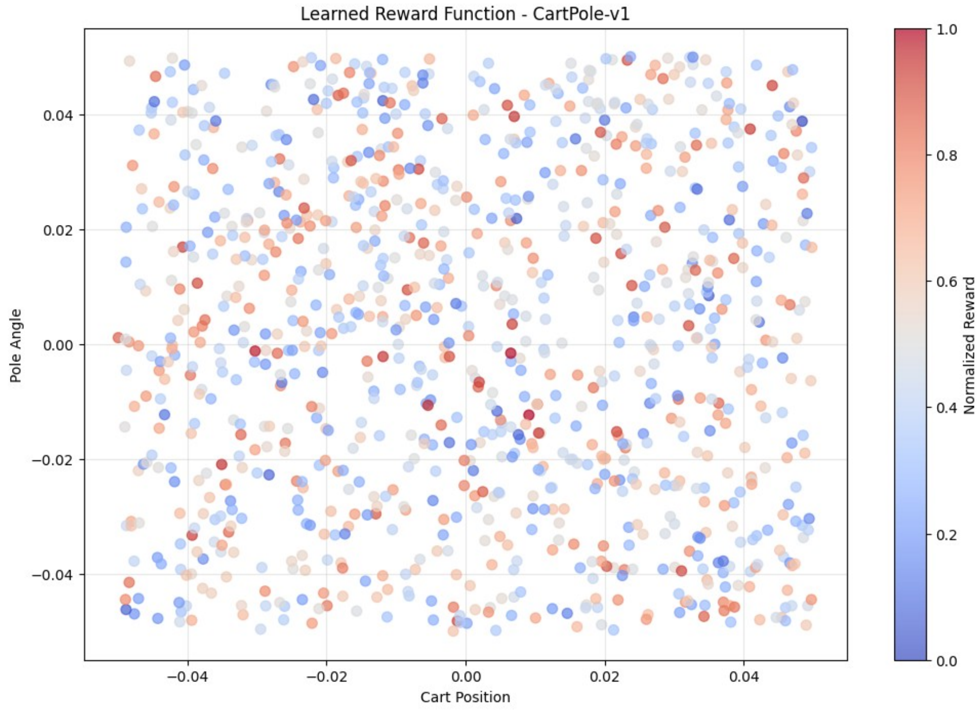
Experiments & Results



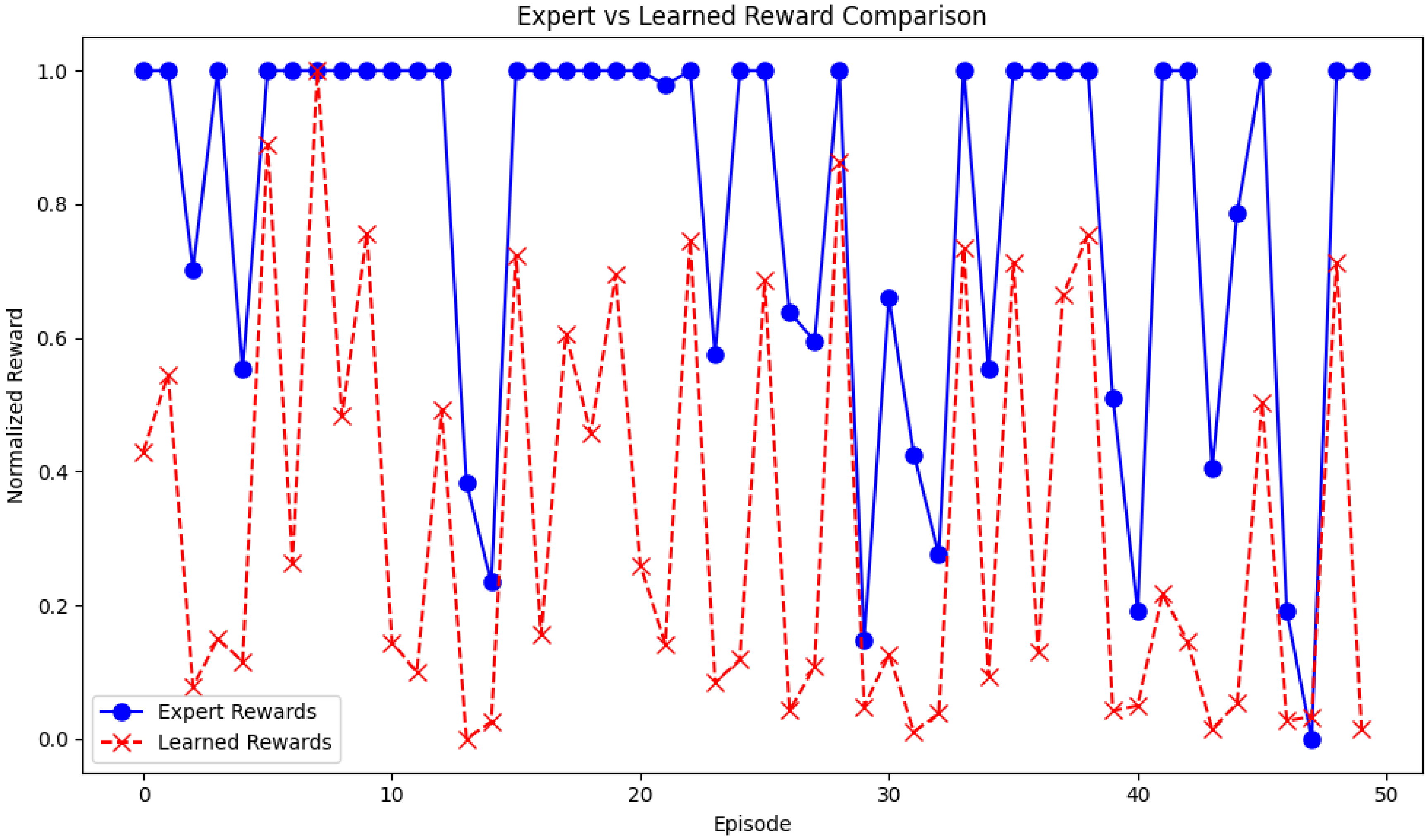
Experiments & Results

Cartpole

- Suboptimal expert
- Learned reward is also suboptimal
- Cannot overcome expert
- Per_cluster width



Experiments & Results



Policy Evaluation

- Evaluation Metrics:
  - Average episode reward (original vs. IRL policy).
- Results:
  - IRL policy matches expert performance in Pendulum env(Trained with PPO for 20m timesteps).
  - Original Score: -191 → IRL Score: -174
  - IRL reward for cartpole matches suboptimal expert (Trained with PPO for 10k timesteps) to a good degree, also showcasing one of the flaws of IRL, which is that it cannot outperform the expert if the expert itself is poor but can lead to better generalizations.

Key Innovations

1. RBF kernels instead of polynomial kernels to catch more non-linear features
2. Automatic hyperparamter tuning with multi-arm bandits to choose K for K-means
3. Adaptive Kernel Width to maximize reward matching
4. L2 regulirization to prevent overfitting

Individual Contributions

Ashiq (CS22B2021) - RBF Networks

- Replaced polynomial features with RBFs for non-linear rewards
- Implemented automated center initialization (vs manual grid search)
- Demonstrated consistent performance gains in Pendulum/CartPole

Ajmal (CS22B2046) – MultiArm Bandits

- Bandit algorithm to auto-select RBF centers (optimal K=2-3)
- K-means + silhouette scores for center evaluation
- Showed faster convergence compared to baseline methods

Abishek (CS22B2054) – Adaptive Width

- 4 adaptive width methods (Density/Cluster/Covariance/Learned)
- Learned widths achieved best performance by silhouette metric
- Resolved fixed-width oversmoothing issues in test environments