The Power of Data: Transforming and Optimizing Representation Space

--Embedding, Interactive Intelligence, and User Profiling

Yanjie Fu

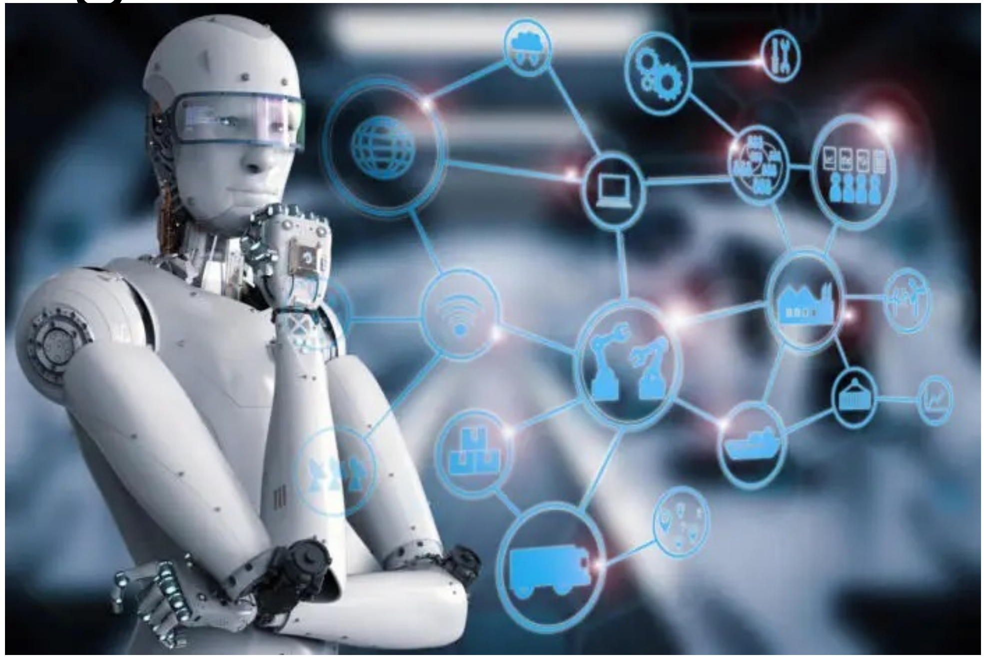
Department of Computer Science

University of Central Florida

About Me

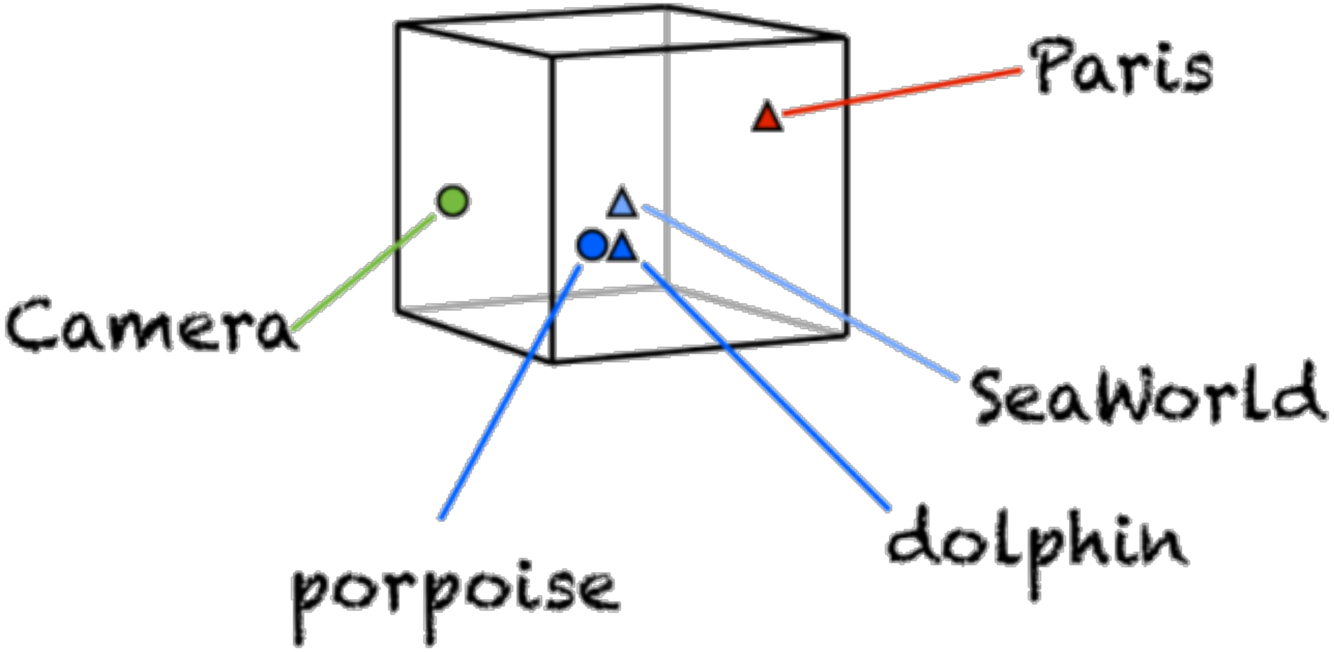
- Assistant Professor of Computer Science, UCF
  - B.E.: University of Science and Technology of China, 2008
  - M.E.: Chinese Academy of Sciences, 2011
  - Ph.D.: Rutgers University, 2016
- Research: representation, automation, interaction of AI systems
  - NSF CAREER Award
  - 4 Best Paper (Finalist) Awards: KAIS Best of IEEE ICDM 2021, ACM TSAS Best of SIGSpatial2020, ACM TKDD Best of SIGKDD2018, KAIS Best of IEEE ICDM2014
  - 29 CSRANKINGS.ORG top papers, 2800+ citations, h-index: 26
- Teaching
  - 1st PhD Student: Pengyang Wang, TTAP at University of Macau
  - 2nd PhD Student: Kunpeng Liu, incoming TTAP
- Service
  - Area Chairs, Senior PC or TPC members of major AI/DM/ML/DB conferences (e.g., KDD, AAAI, IJCAI, ICML, NeurIPS, ICLR, ICDM, WWW, SIGIR, SDM, ICDE, VLDB); Guest Editor of ACM TIST special issue on deep spatiotemporal learning

Ultimate goals of AI: machine intelligence = human intelligence



- System 1 intelligence: **representation** (what happened) and projection (what will happen)
- System 2 intelligence: reasoning (how and why it changes) and decision (how to change it)

A Representation (Feature) Space



Representation is a fundamental perceptual intelligence and algorithm enabler

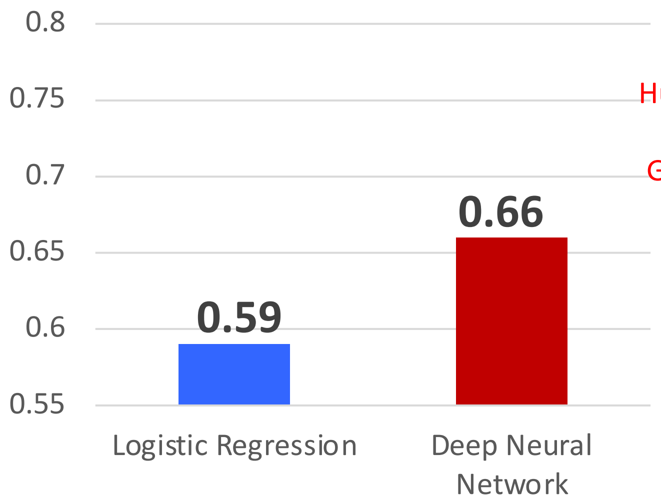
- Provide machines with situation awareness to characterize the state
- Identify and disentangle the unobserved drivers hidden in data to fight against the curse of dimensionality
- Easy the extraction of useful information in predictive modeling
- Reconstruct distance measures to form smooth discriminative patterns
- Automate and eliminate feature engineering
- Many learning tasks such as learning with unlabeled data, small data, or data fusion are built on representation frameworks
- Embed structure knowledge into representation to make classic ML algorithm available to complex data
- Alignment across domain (domain adaptation, distribution shift)

Why an optimal data representation space matters

Task: Identify which customers will make a specific transaction in the future (classification)

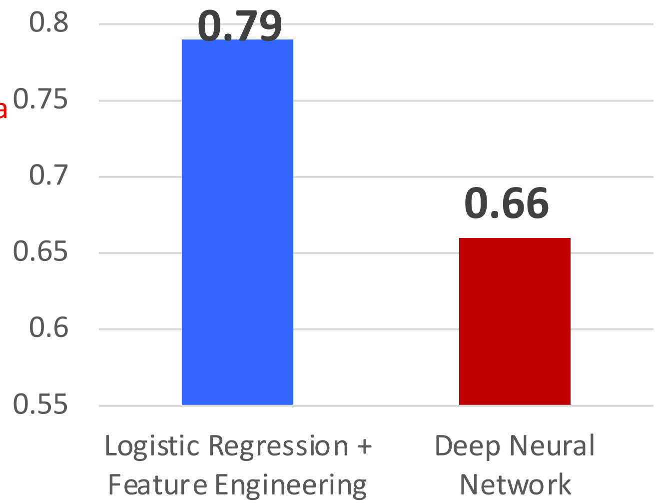
Data:

https://www.kaggle.com/c/santander-customer-transaction-prediction/data



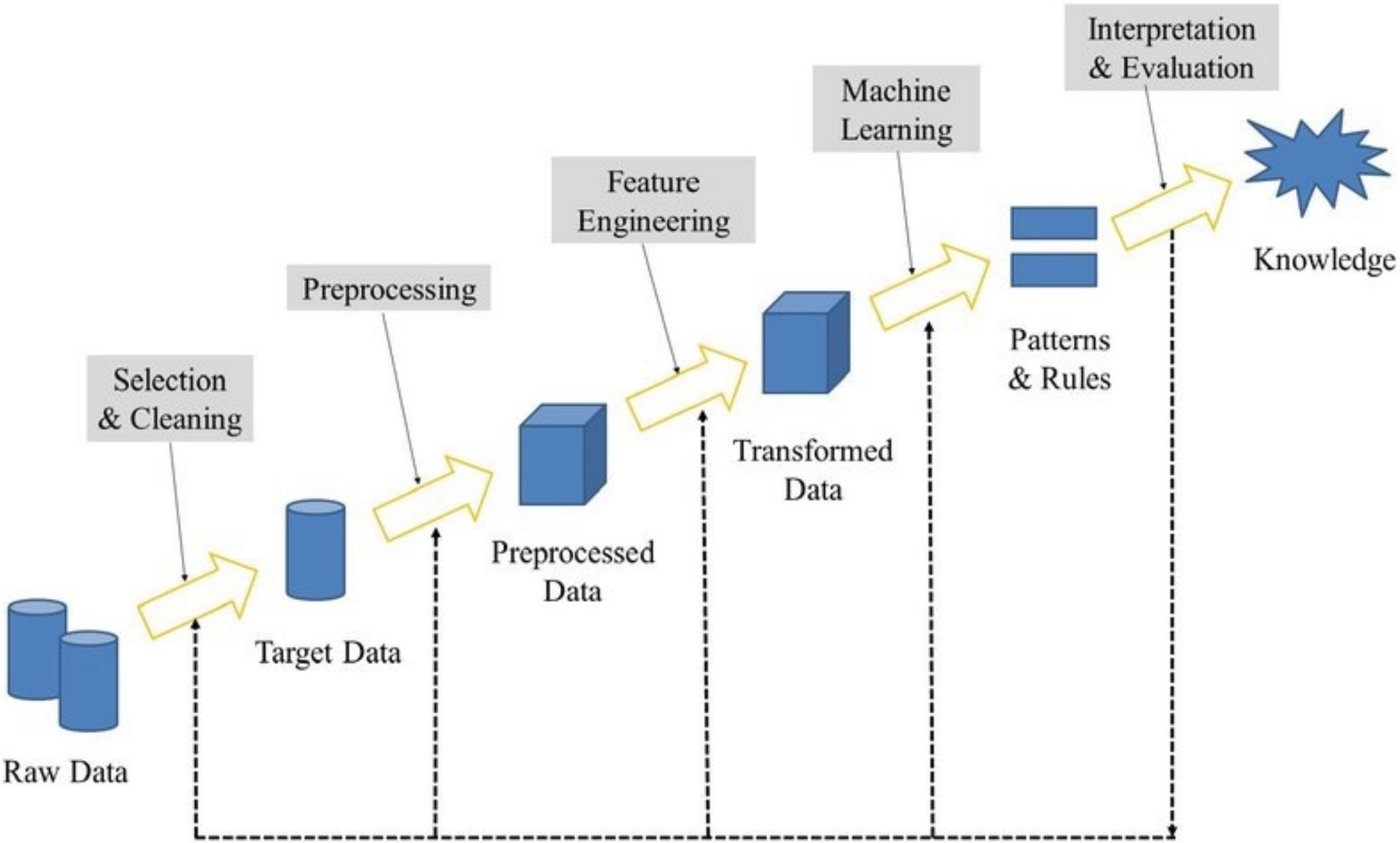
AUC Performance, the **higher** the **better**

Human-Reconstructed Data via Feature Selection, Generation, Preprocessing

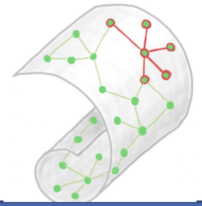


AUC Performance, the **higher** the **better**

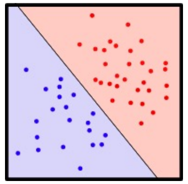
Classic machine learning pipelines



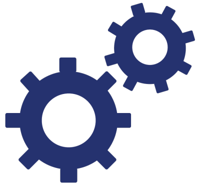
The concept of representation renovates classic ML pipelines



Collect Complex Data



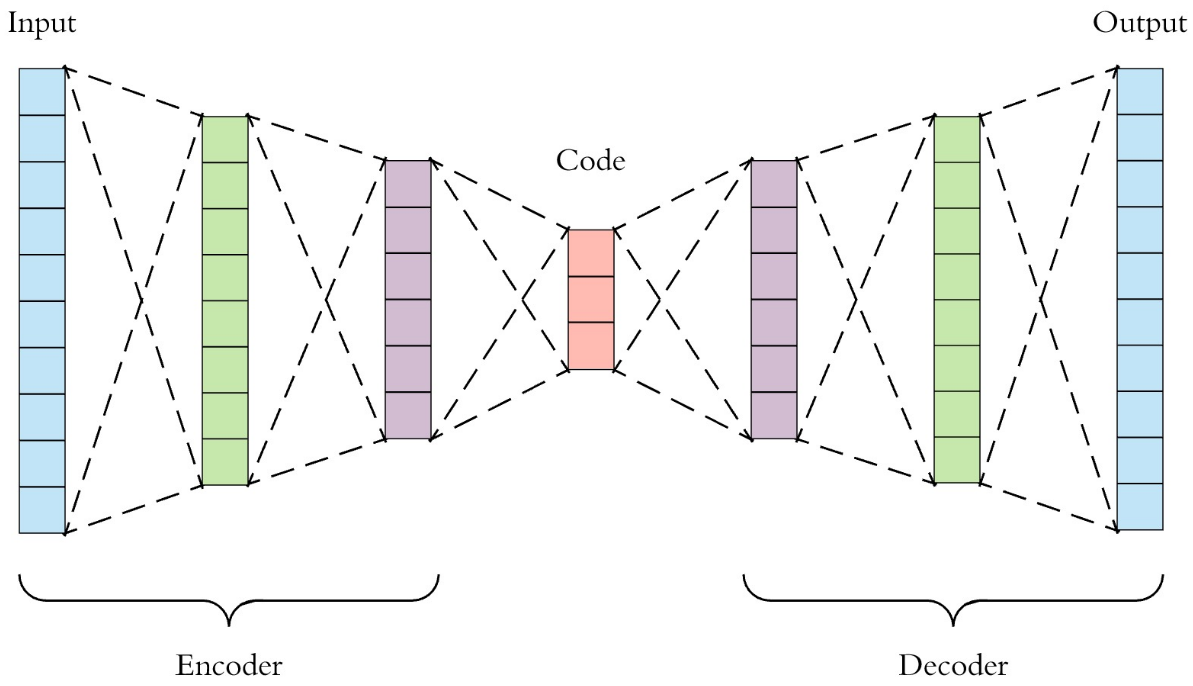
Compute An Optimal Data Representation



Apply ML (Approximation, Error, Training)

Shifting research focus of many studies from optimizing model space to optimizing data space

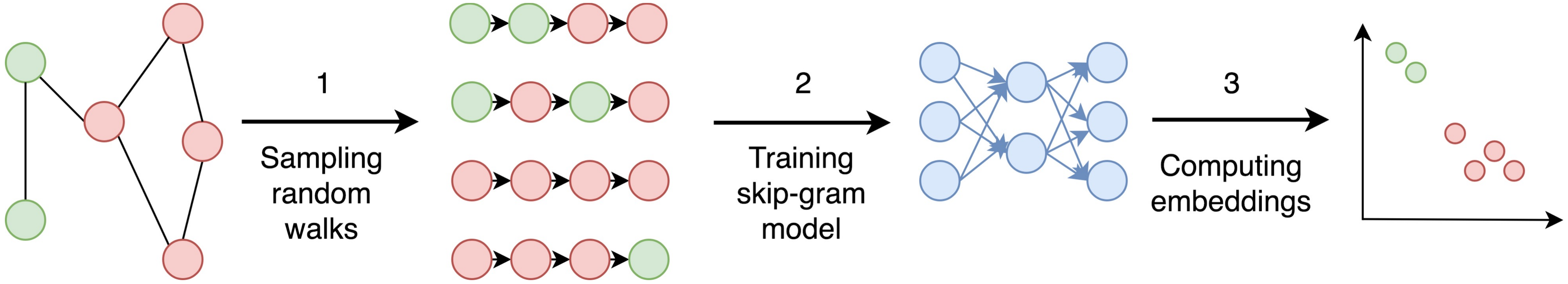
Autoencoder [Hinton et al.]



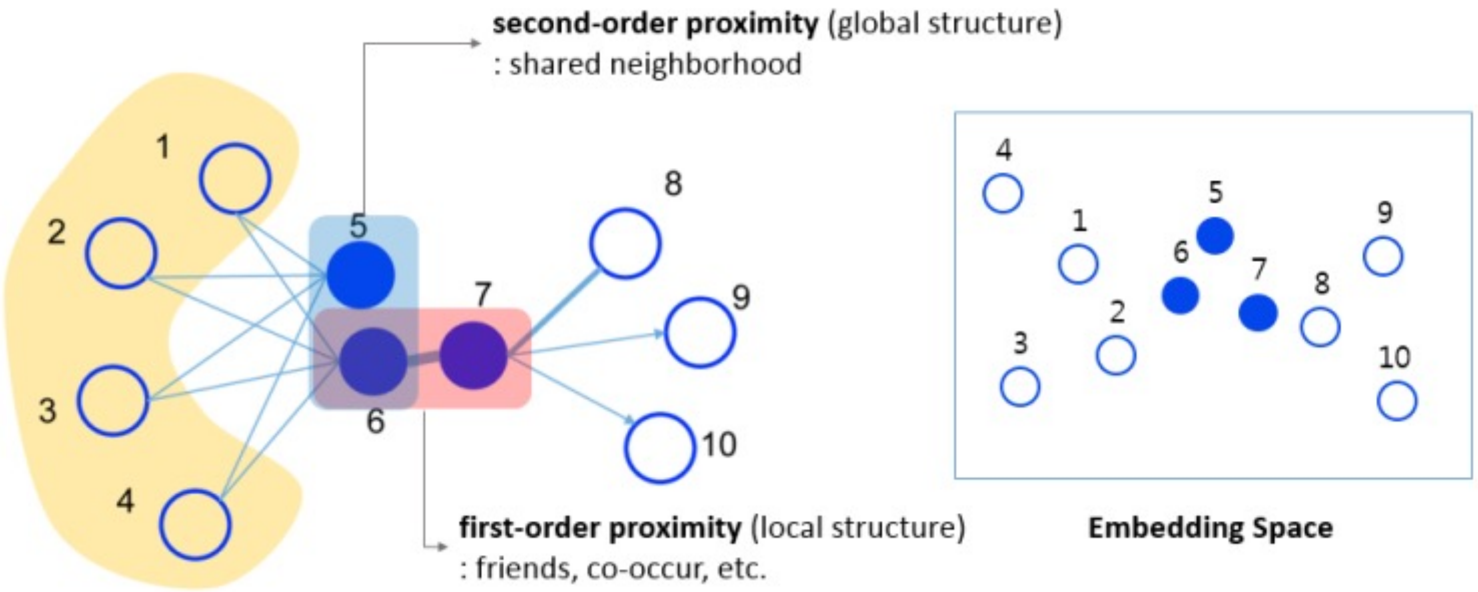
- Input: a graph/matrix; output: embedding of graph/matrix
- The neural encoder and decoder framework
- Goal: Learn embedding by minimizing reconstruction loss

DeepWalk [Perozzi et al.]

- Input: a graph; output: embeddings of nodes
- Goal: Treat *random walks on networks* as sentences and then learn node representations with the Skipgram word embedding
- Empirically produce a low-rank transformation of a network's normalized Laplacian matrix



LINE: Large-scale Information Network Embedding [Tang et al.]



- Inputs: Directed, undirected, or weighted networks
- Goal: learn a node embedding encoder that
  - Preserve the first-order (node-node distance) proximity
  - Preserve the second-order (neighbor-neighbor structure similarity) proximity

Collective representation learning

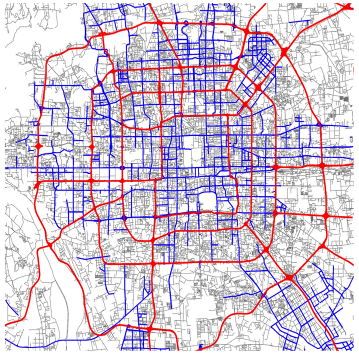
- Learning representations from collectively-related graphs

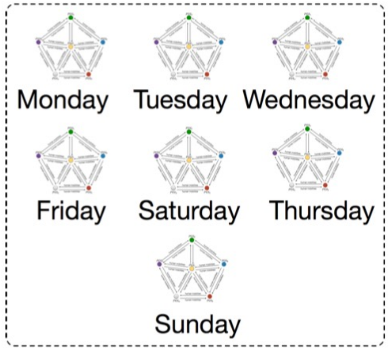
A city

Multiple Graphs

A feature vector

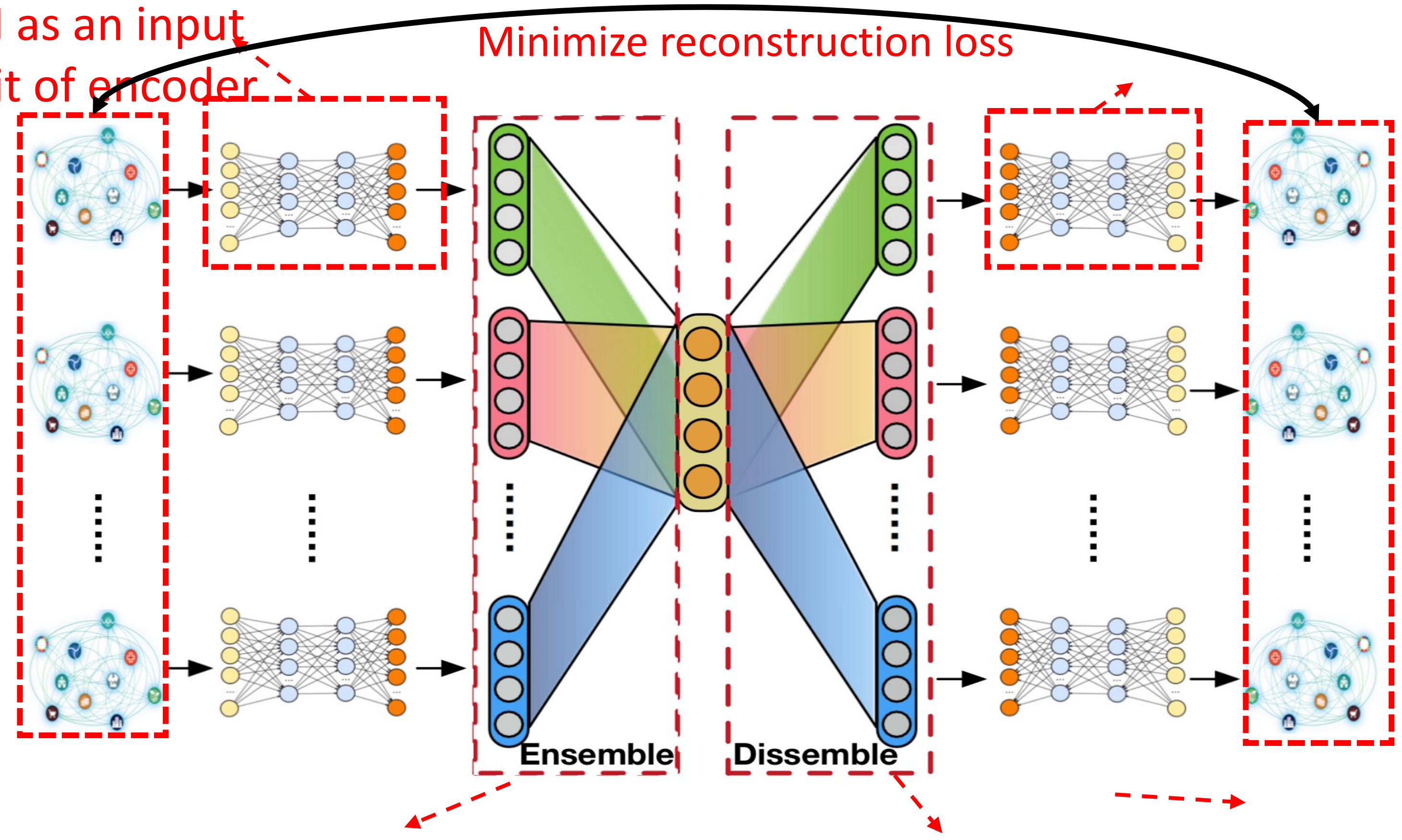
f(





) =

Our Solution: An Ensemble-Encoding Dissemble-Decoding Method



signal ensemble (Multi-perceptron summation)

signal dissemble (Multi-perceptron filtering)

A Collective Embedding Perspective with Periodic Spatial-temporal Mobility Graphs. ACM Transactions on Intelligent Systems and Technology, 9, no. 6 (2018): 63

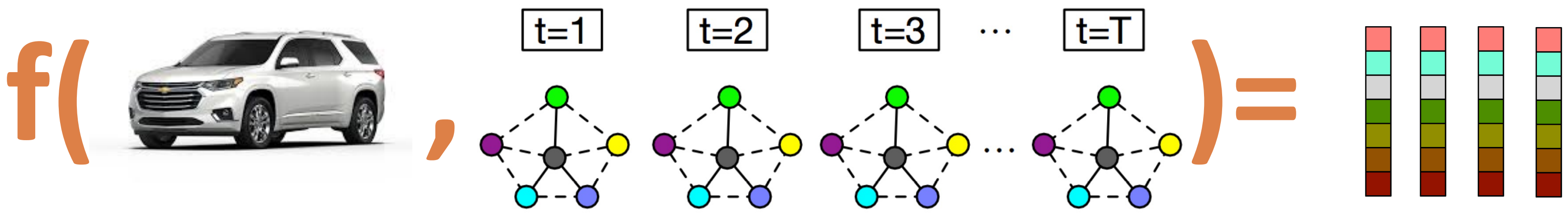
Dynamic representation learning

- Temporally-related graph streams

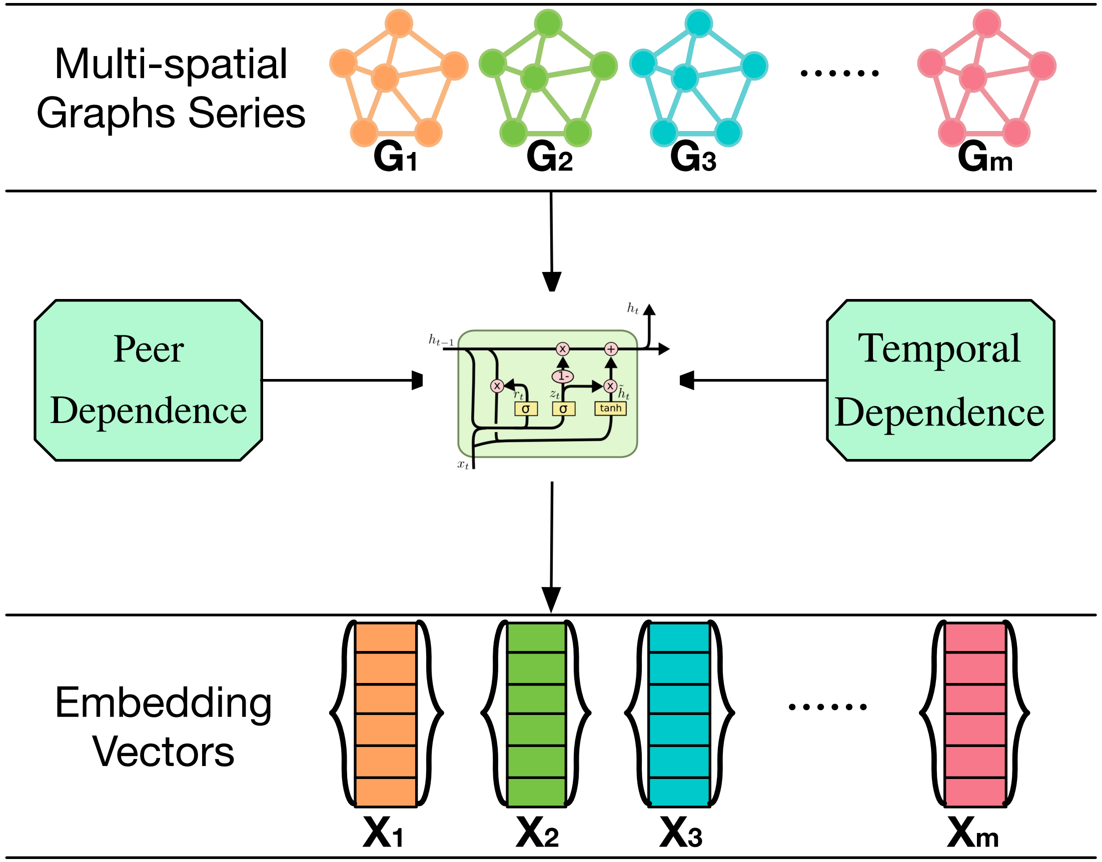
A car

Graph stream

A sequence of feature vectors (driver profiles)



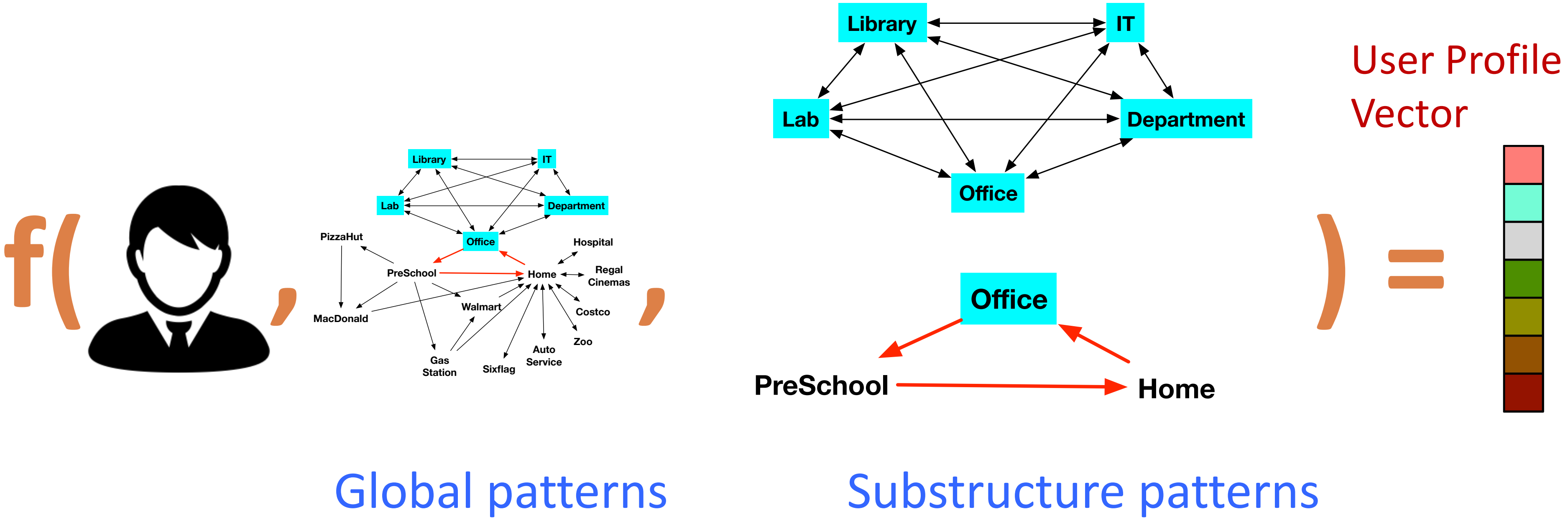
Our Solution: Peer and Temporal Gated Recurrent Unit Encoding



You Are How You Drive: Peer and Temporal-Aware Representation Learning for Driving Behavior Analysis. (KDD18)

Substructured representation learning

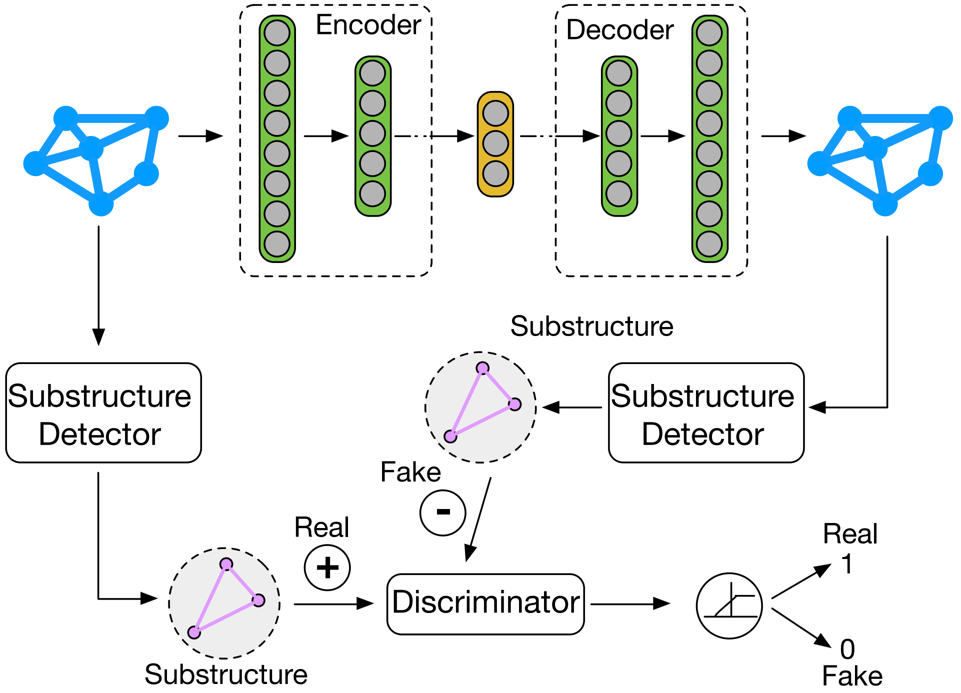
- A globally-structured graph with unique subgraph patterns



Substructured Representation Learning: Learning the feature representation of a graph with attention to preserving unique subgraph patterns

Our Solution: Adversarial Substructured Learning

- **Preserving global structure:** minimizing the reconstruction loss between input graph and reconstructed graph
- **Preserving substructure:** use adversarial training to force deep encoder-decoder to pay attention to subgraphs

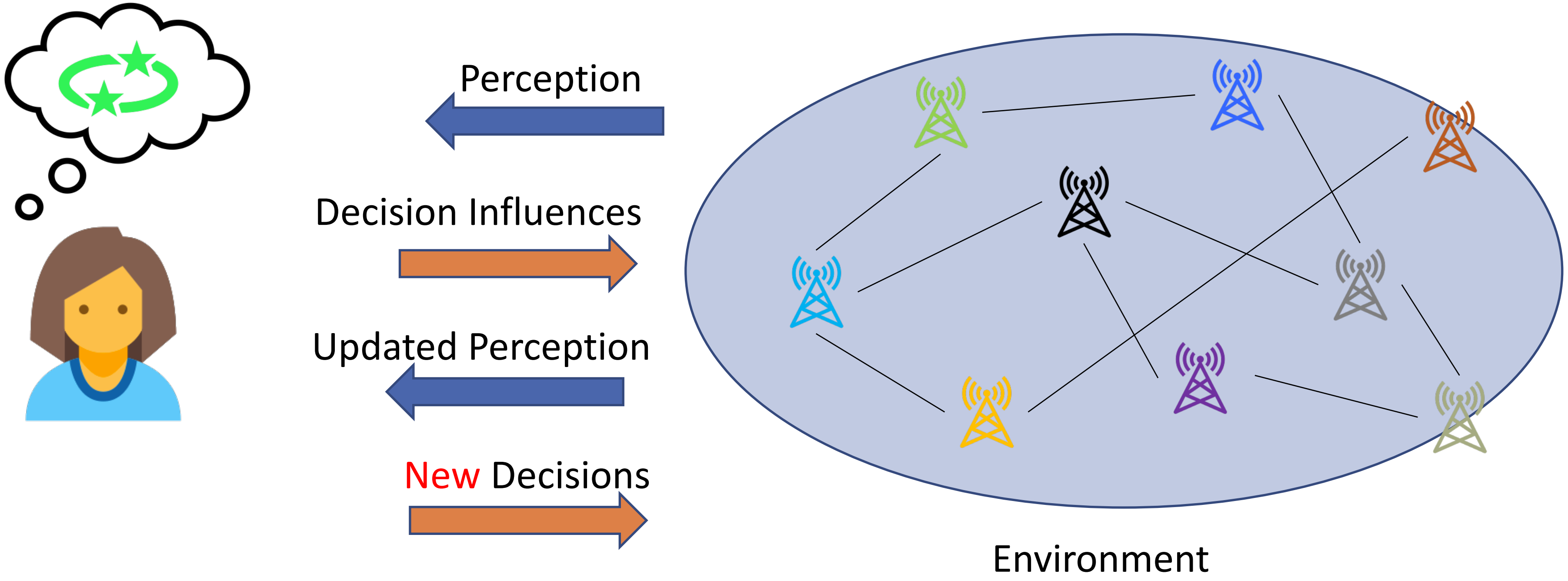


Adversarial Substructured Representation Learning for Mobile User Profiling. (KDD19)

Human-environment interaction: interactive perception and decision

Human Perceives the Environment

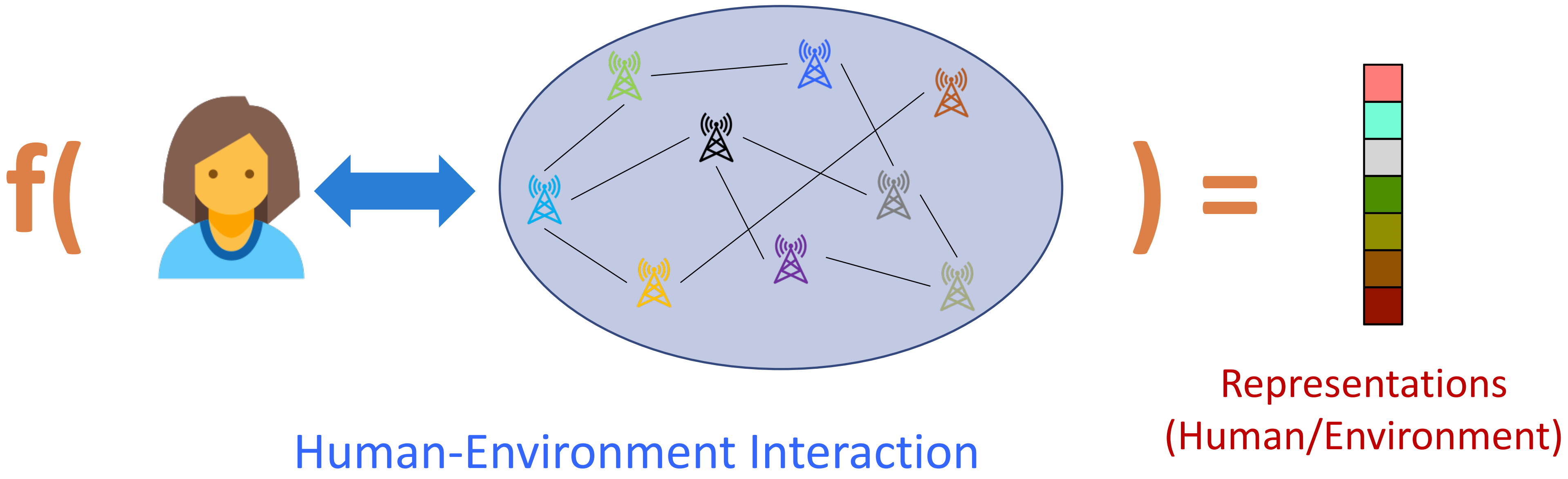
Human's Decisions Change the Environment



Human Has New Perception on the Updated Environment

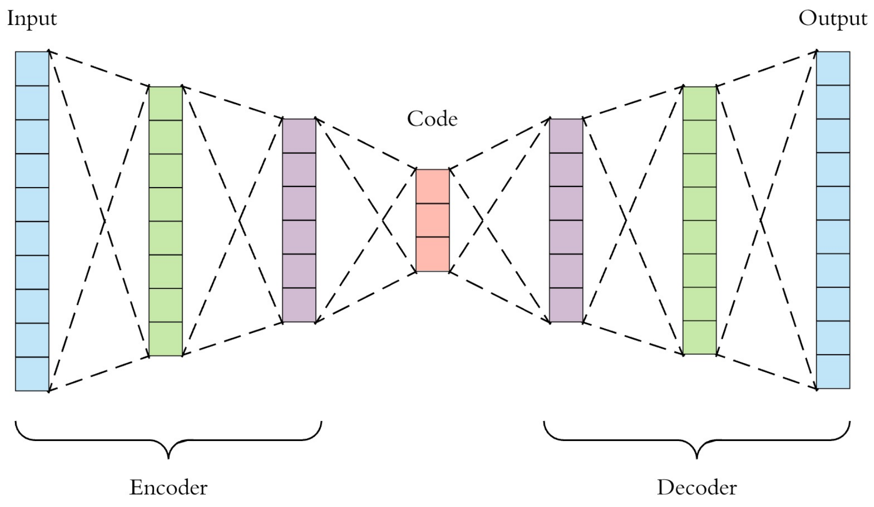
Human Makes New Decisions Based on the New Perception

Interactive representation learning



Expectation: Representations are updated incrementally along with interactions

Traditional representation learning criteria cannot model interactions



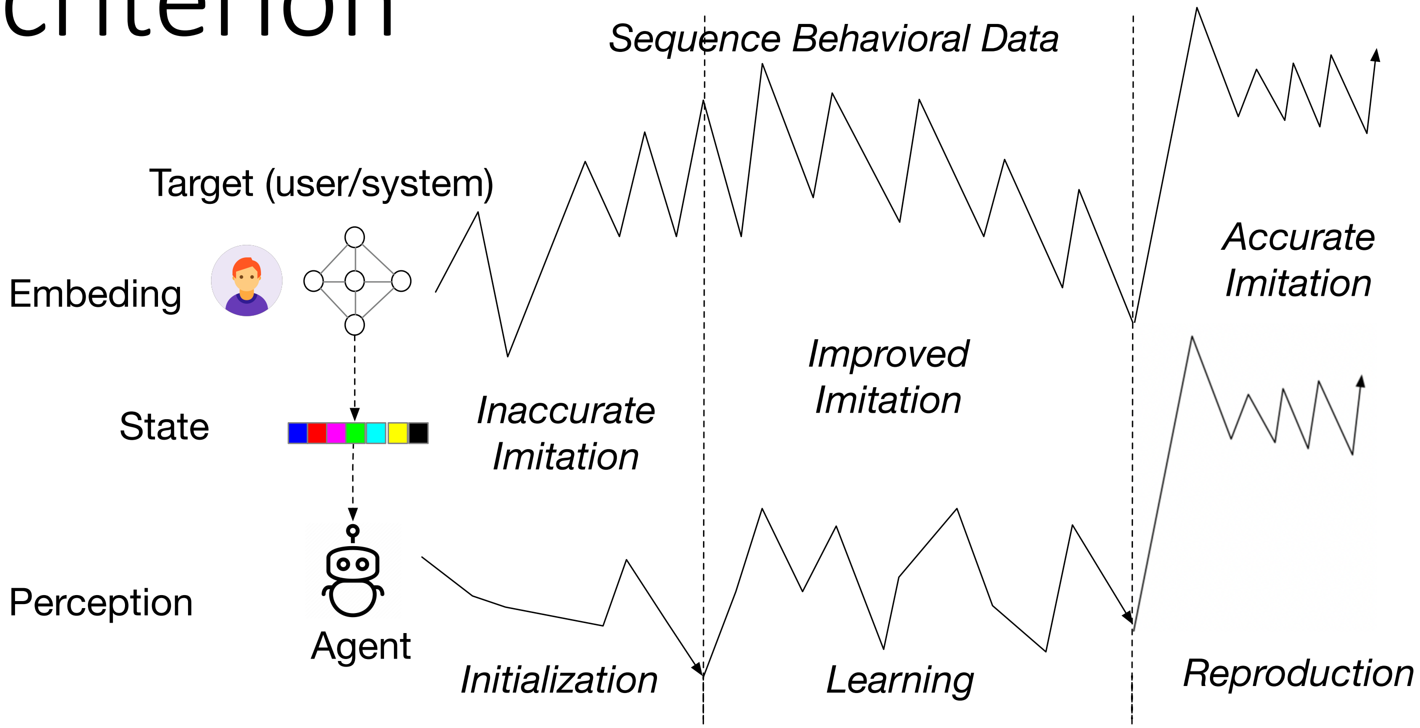
Reconstruction Loss

Minimizing the reconstruction loss between the input and reconstructed output

Downstream Task-Oriented Loss

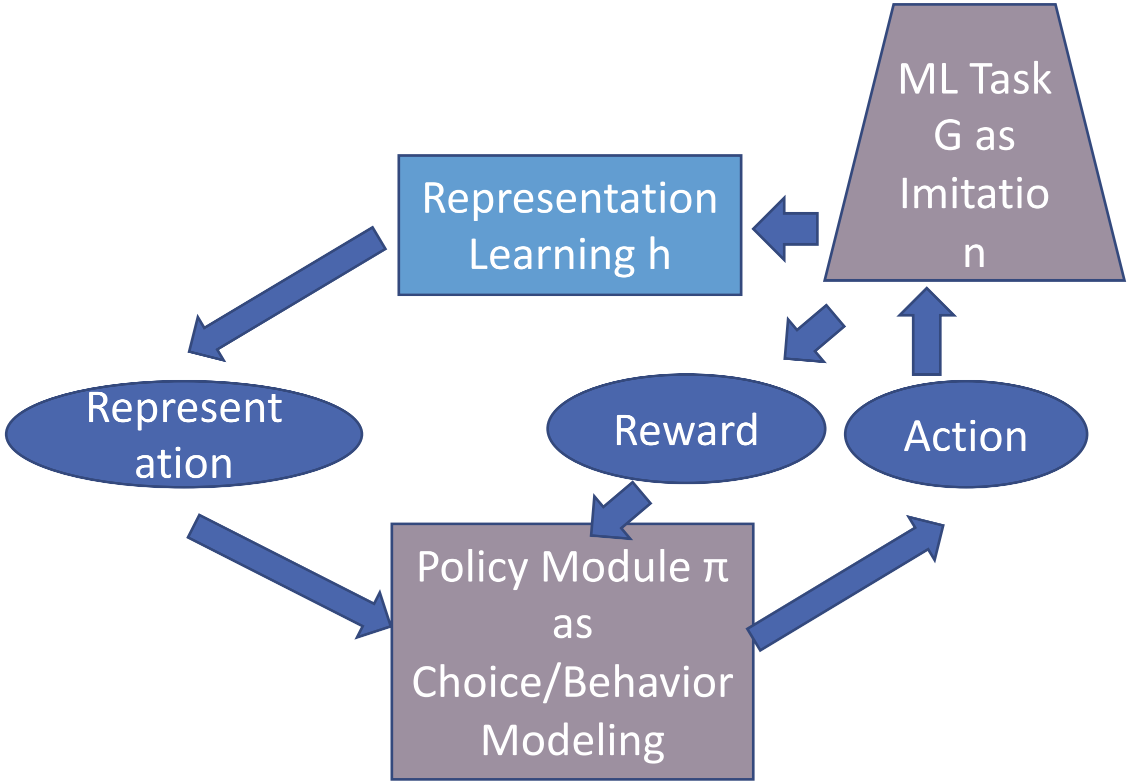
Minimizing the prediction loss

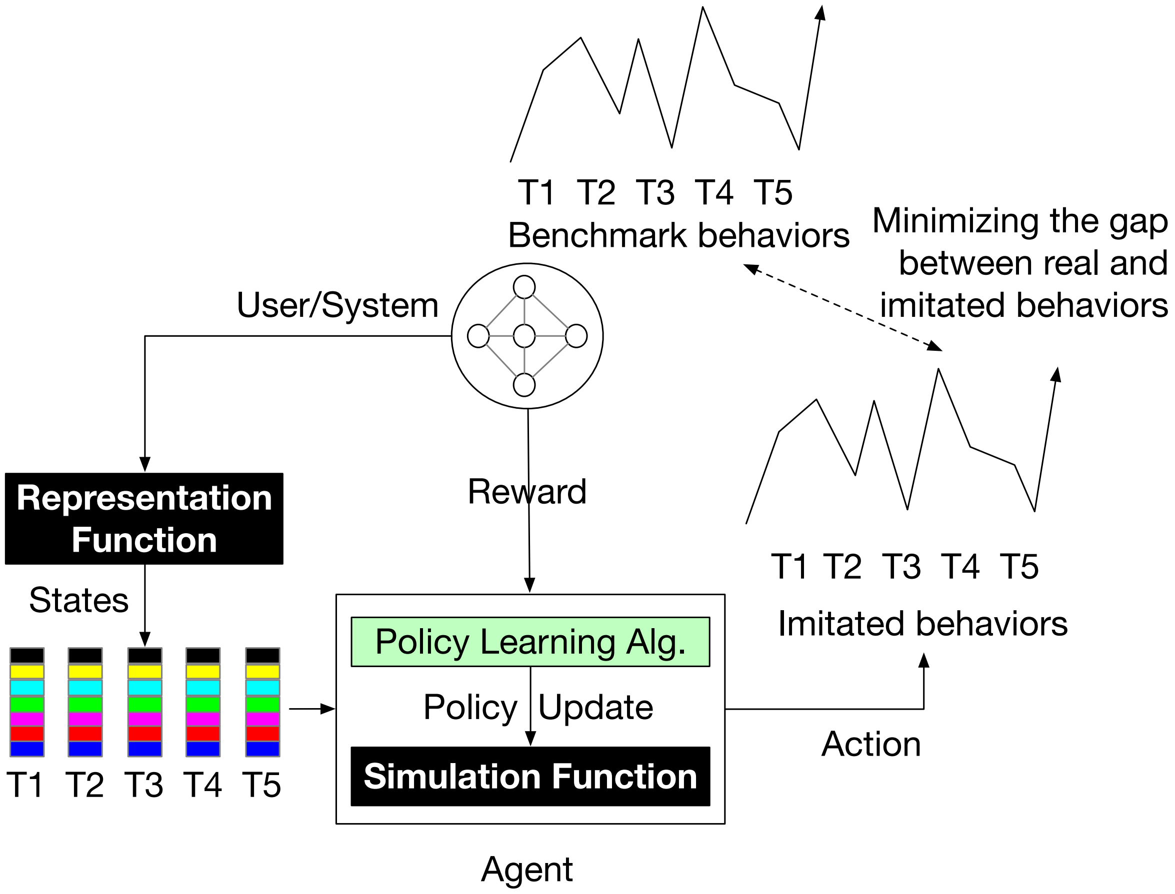
A new representation criterion: imitation-based criterion



- Suppose a user/system is perfect in understanding and per the environment
- Train an agent to simulate (mimic) human's behavior based on the learned representations of the environment
- The learned representations (perception) is considered perfectly, once the agent can copy human's behavior patterns

Our framework: reinforcement interactive representation learning

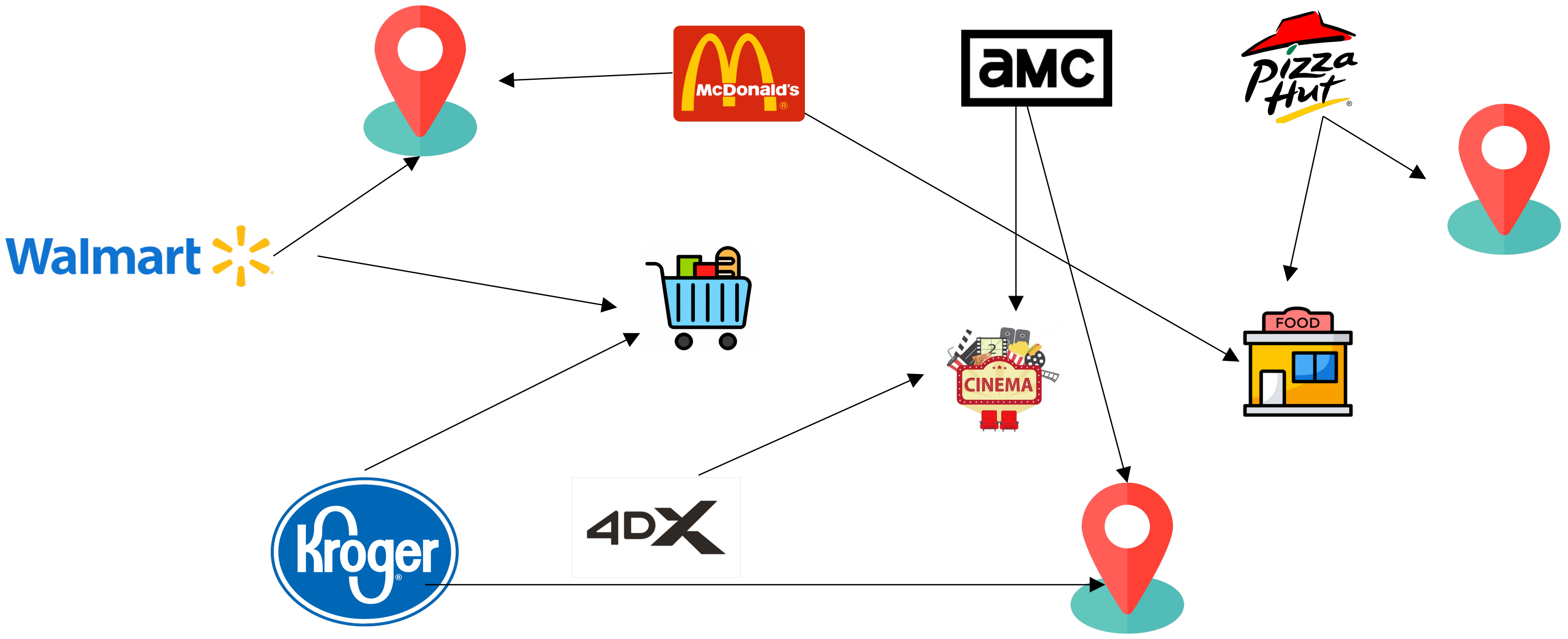




"Incremental mobile user profiling: Reinforcement learning with spatial knowledge graph for modeling event streams." (KDD20)

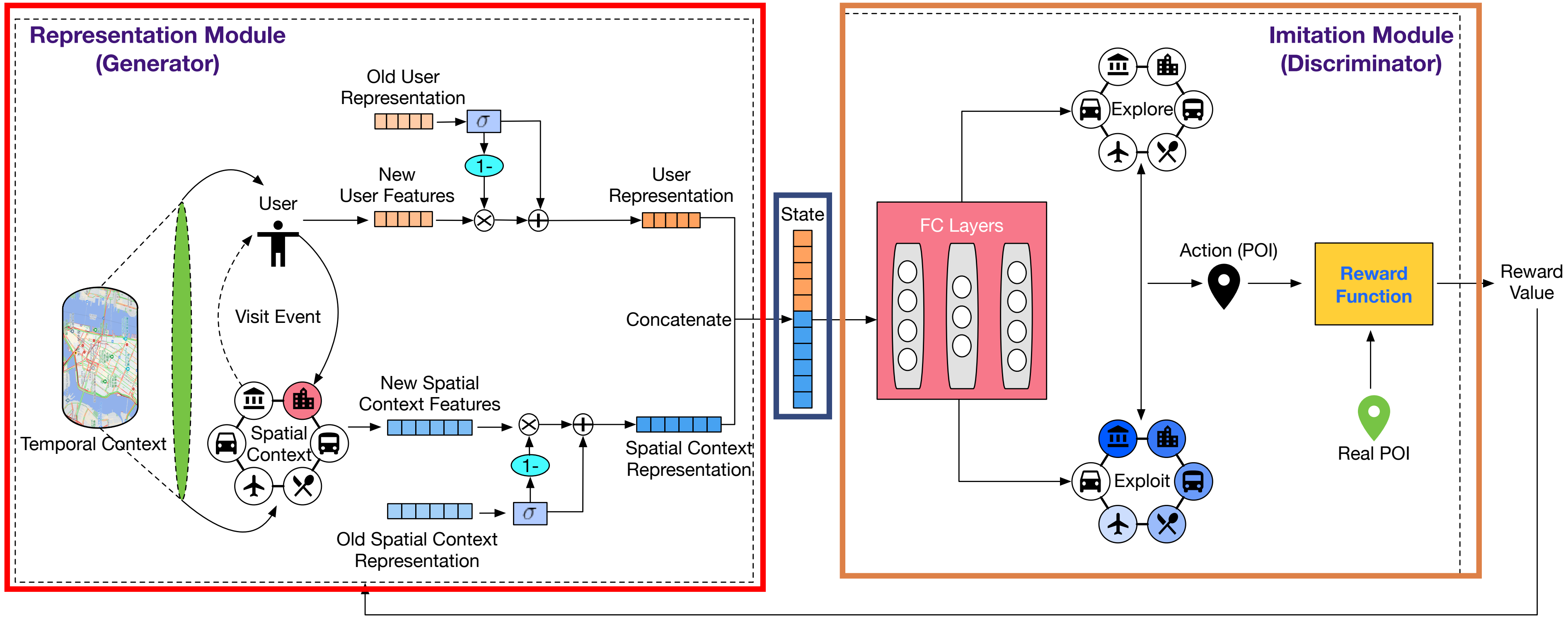
A concrete example: what to do next - inferring next activity

- Spatial Knowledge Graph as the Physical Environment

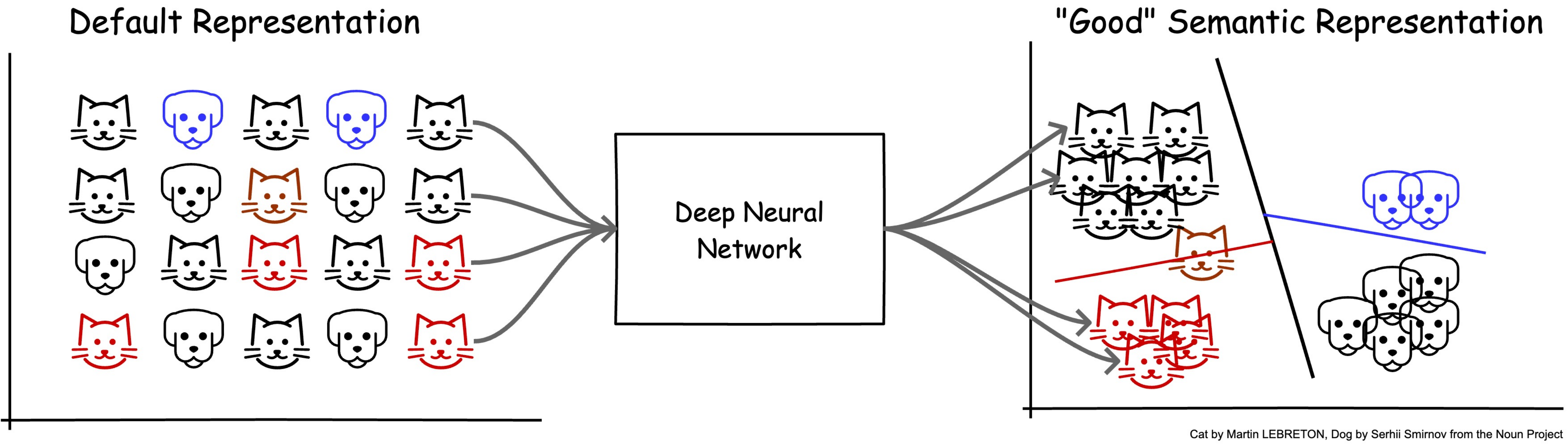


- Entities: POIs, POI categories, functional zones
- Relations: Locate at, Belong to
- Facts:
  - <POI, “belong to”, POI category>
  - <POI, “locate at”, functional zones>

Model components and structure

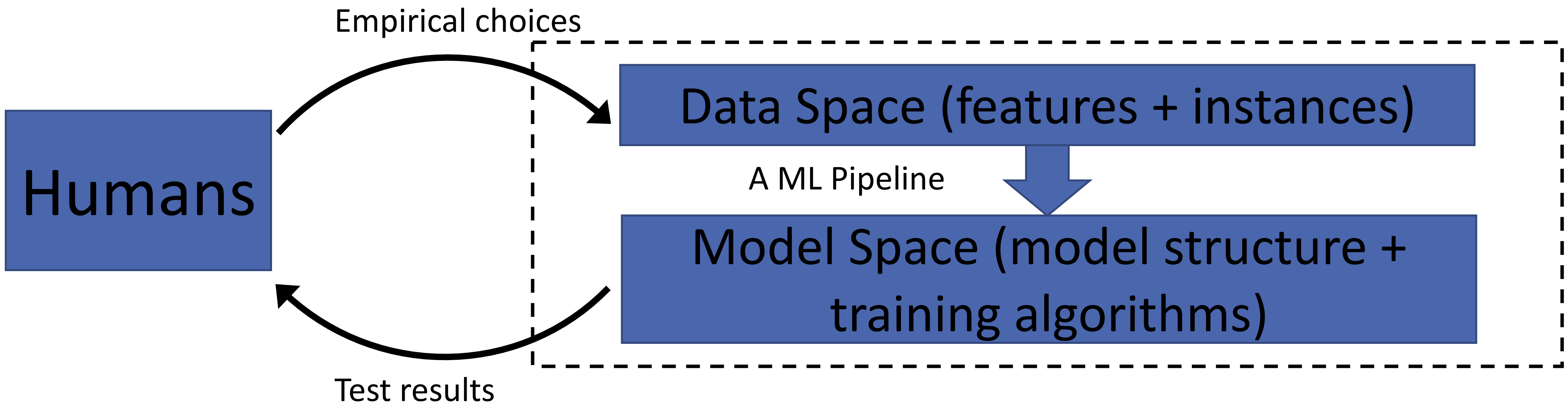


Research gap: how deep AI optimizes data representation?



- Automated
- Latent, black-box, unexplainable

Research gap: how humans optimize data representation?



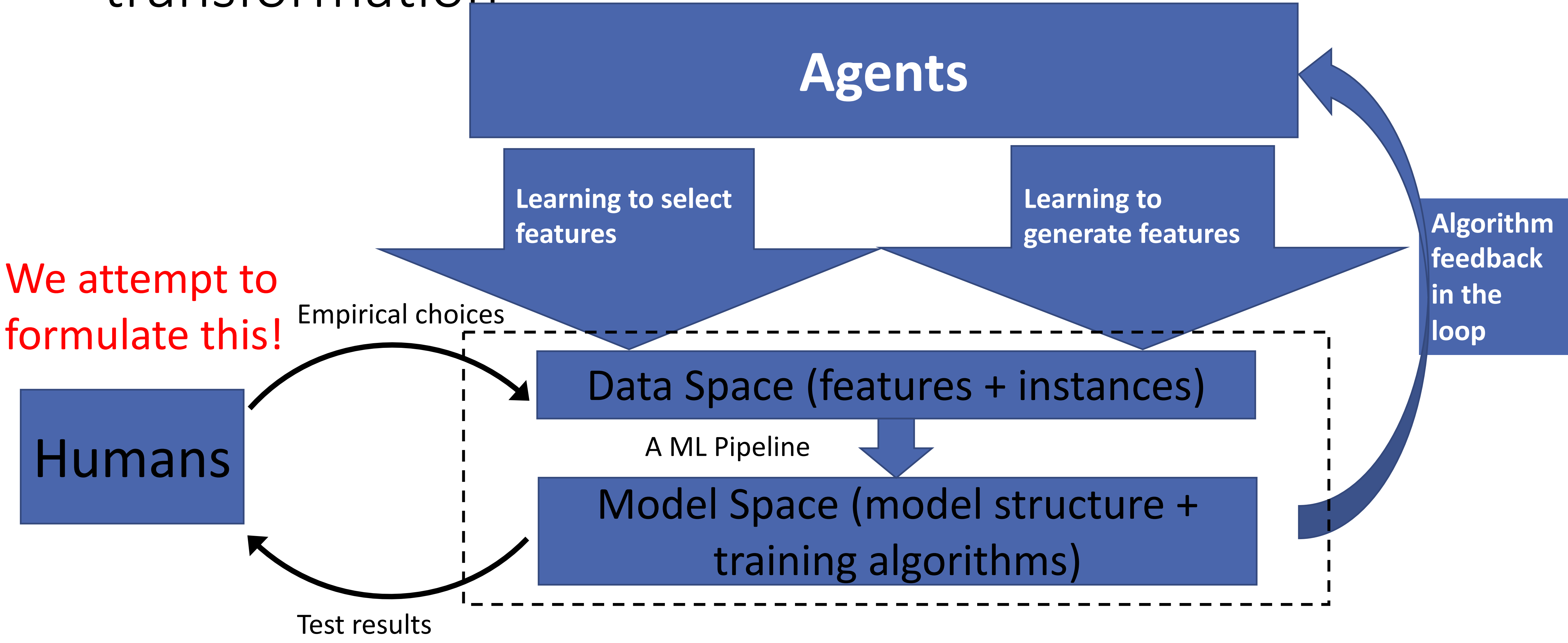
- Explicit and explainable
- Time-consuming, brittle, incomplete

Reimagining the future of representation (feature) space

Can we equip representation intelligence with full automation, explainable explicitness, flexible optimal?

- Full automation: how can we make ML less dependent on feature engineering, construct ML systems faster, and increase applicability of ML?
- Explainable explicitness: how can we ensure traceable and explainable explicitness in reconstructing data transformation?
- Flexible optimal: how can we create a framework to reconstruct a new data transformation for any given predictor?

Self-optimizing Data Geometry: Learning to reconstruct an optimal and explainable data transformation



Research overview of this project

	Feature Selection	Feature Generation
Explicit Reconstruction	Multi-agent Reinforcement Feature Selection **(KDD19, TKDE-a)**	Hierarchical Reinforcement Feature Generation **(TOIS)**
	Addressing Learning Efficiency **(ICDM22 Best Paper Finalist, ICDM20)**	Group-wise Cascading Reinforcement Feature Generation **(KDD22)**
	Interactive Learning Mechanisms **(ICDM20, ICDM21, SDM21, TKDE-b, KAIS)**	
	Multi-arm Bandit Feature Selection **(SDM21)**	

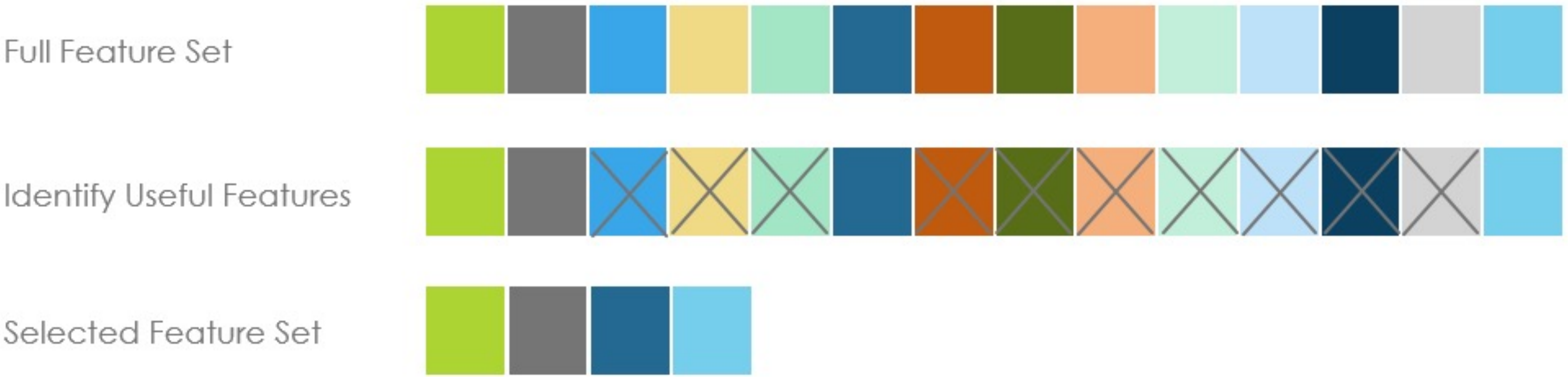
Implicit Learning

- Collective Representation Learning **(TIST)**
- Substructured Aversarial Representation Learning **(KDD19, TKDE-d)**
- Interactive Representation Learning **(KDD20, TKDE-e)**
- Dynamic Representation Learning **(KDD18, TKDE-c)**

Explainable and Optimal Representation Space Reconstruction: A Selection Perspective

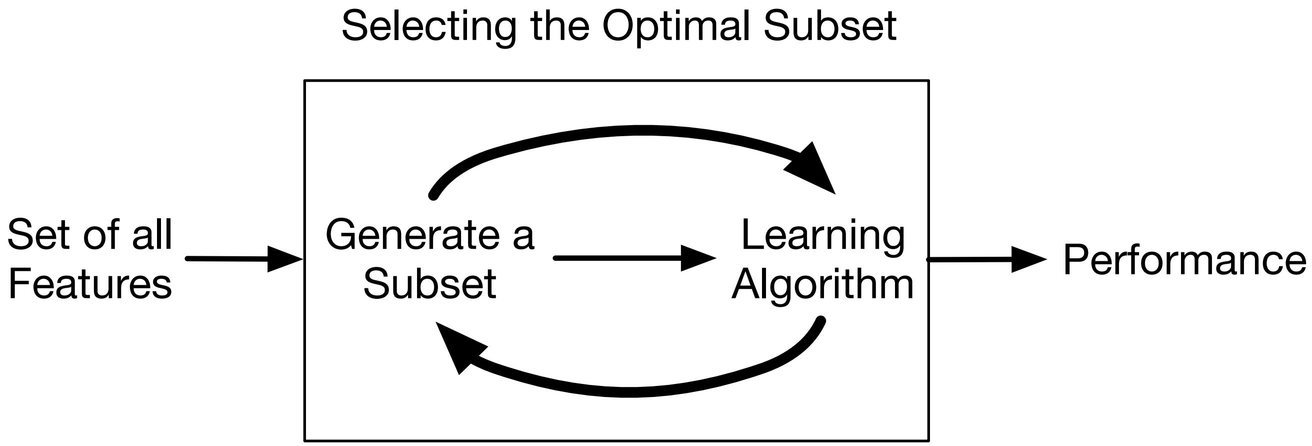
(KDD19, TKDE-a, TKDE-b)

Feature selection

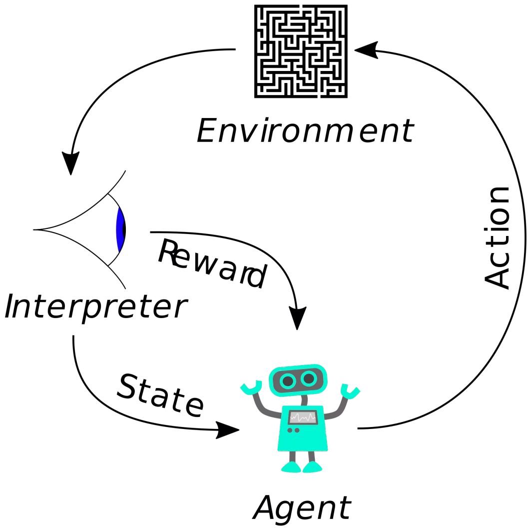


Feature selection as an exploration process

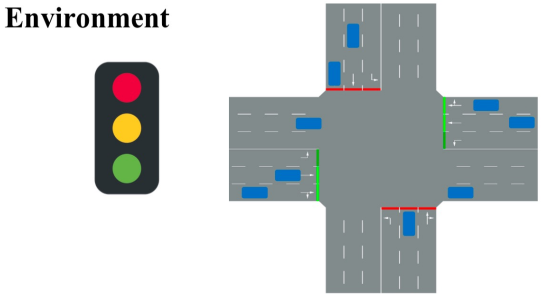
- Feature selection: an iterative exploration process to find an optimal / near optimal subset/subspace of features



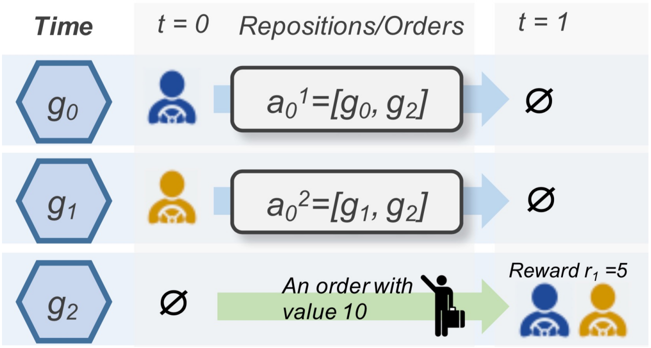
Reinforcement learning as a tool of exploitation and exploration



- Applications:



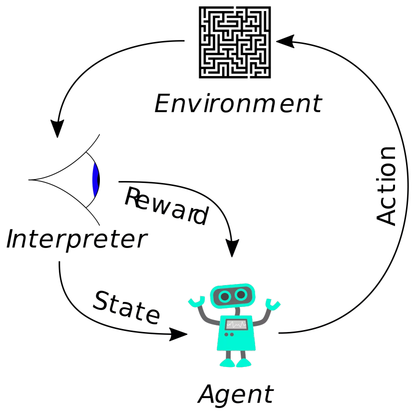
Traffic light control via RL



Taxi fleet management via RL

Automated feature subspace exploration

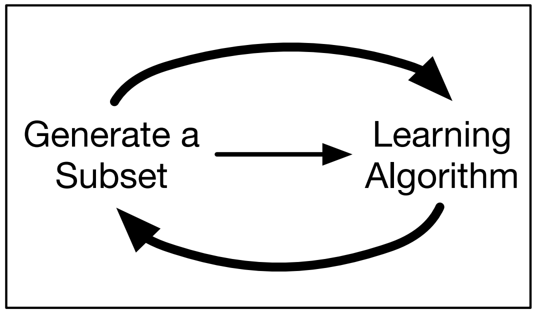
- Inspiration: Can reinforcement learning help to automate feature selection?



Reinforcement Learning
As A Tool of Exploration

+

Selecting the Optimal Subset



Feature Selection
(Exploration Problem)

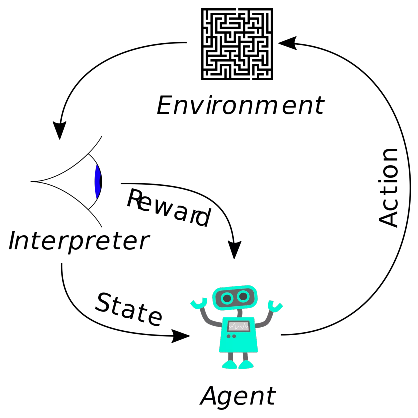
= ?

Traceable & explainable + automated & self learning + global optimal

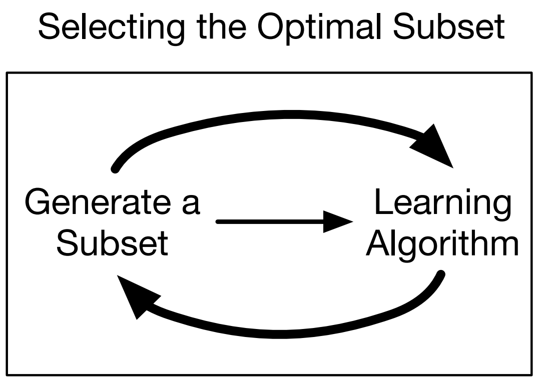
Overview of RFSL for explainable and optimal representation subspace reconstruction

Reinforcement feature selection learning (RFSL): learn a feature selector that

- Traceable: record selection process and understand semantic feature labels
- Self-optimizing: automatically automatically select the best feature subset to identify an optimal representation subspace



Reinforcement Learning
As A Tool of Exploration

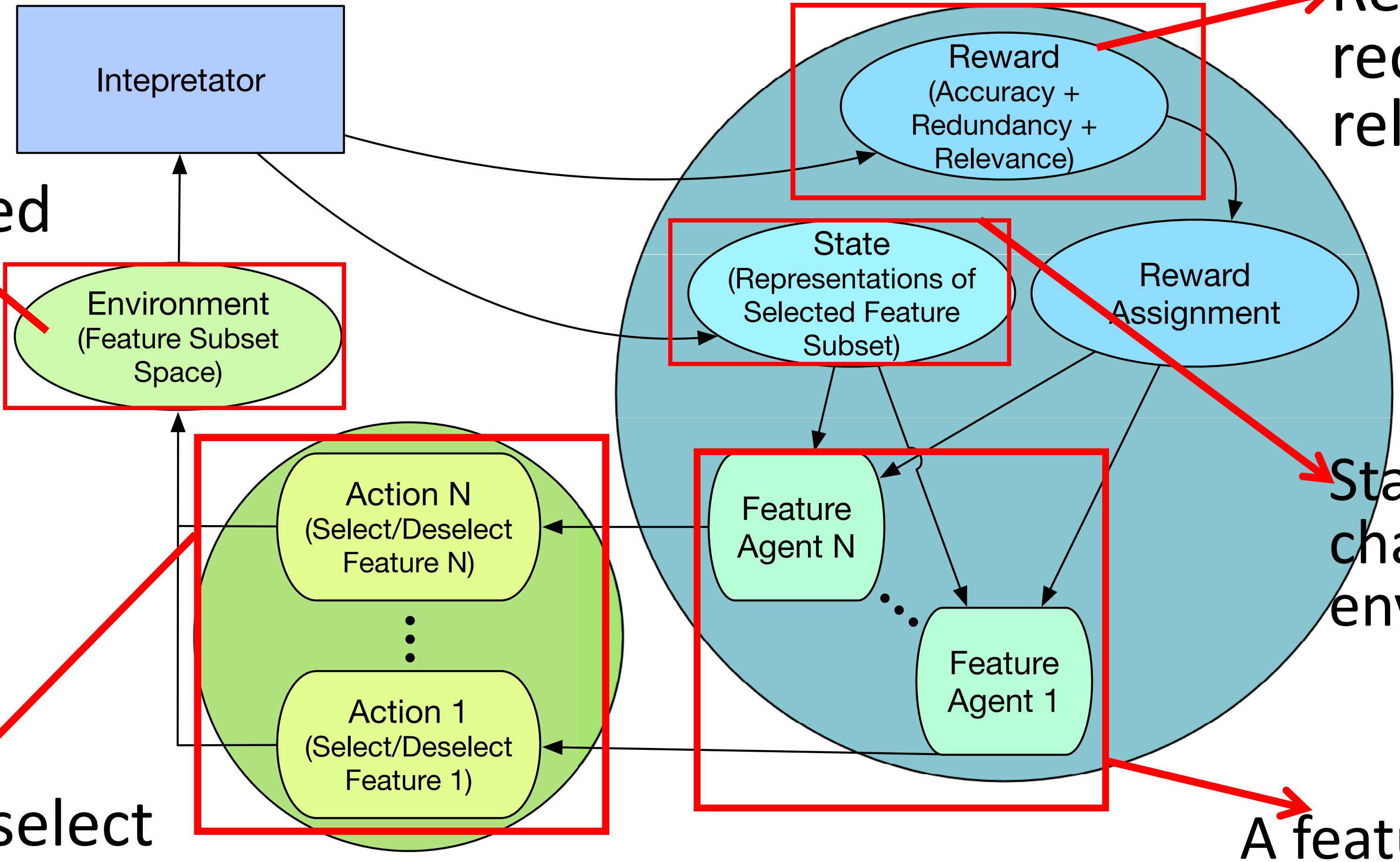


Feature Selection
(Exploration Problem)

Our goal: leveraging reinforcement intelligence for self-optimizing selection

Env is the selected feature subset

Reward: accuracy + redundancy + relevancy

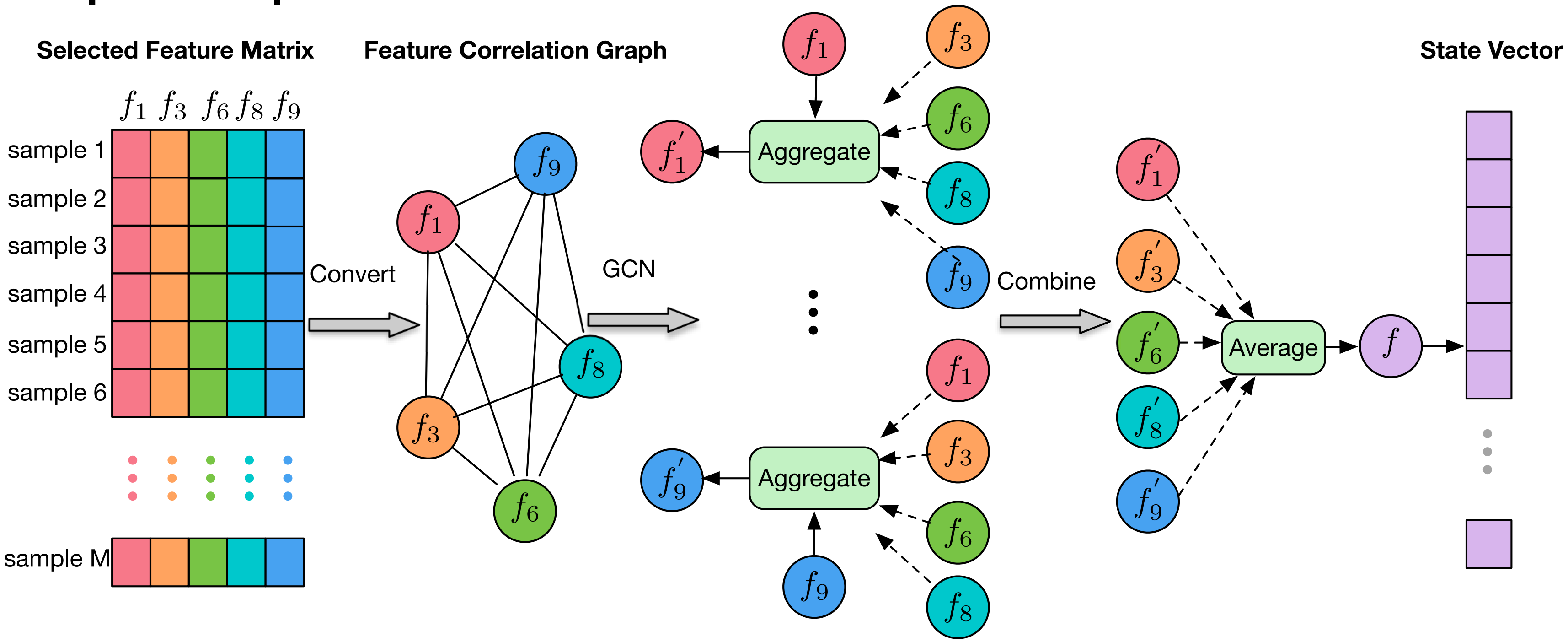


State: the situational characteristics of the environment

An action is to select or deselect

A feature is controlled by an agent

How to accurately represent situational state: a graph perspective

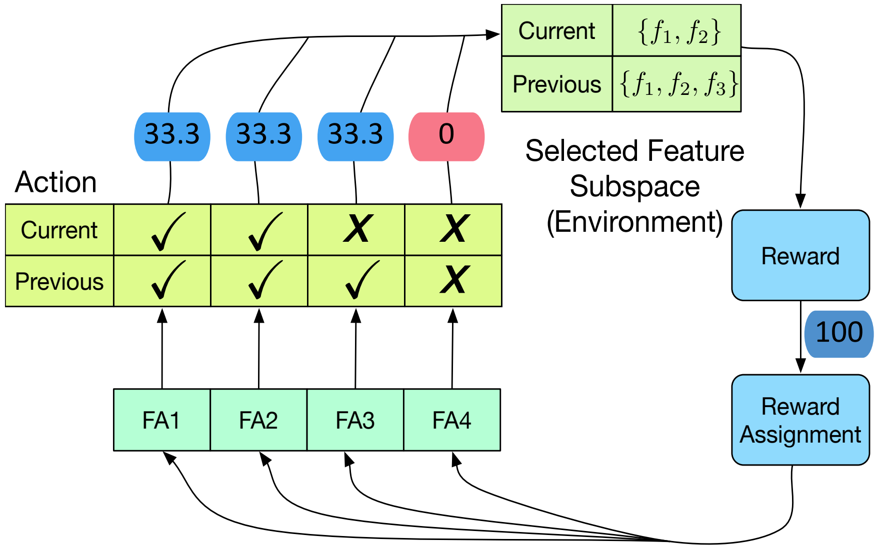


- Step 1: Draw a fully-connected feature-feature correlation graph.
- Step 2: Update each feature's representation.
- Step 3: Aggregate all features' representations.

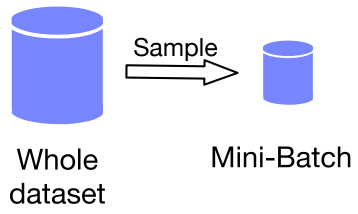
How to fairly assign reward: personalize incentives for participating and non-participating agents

	Current Iteration	Previous Iteration
Participating agents	Select	Select
	Select	Deselect
	Deselect	Select
Non-participating agents	Deselect	Deselect

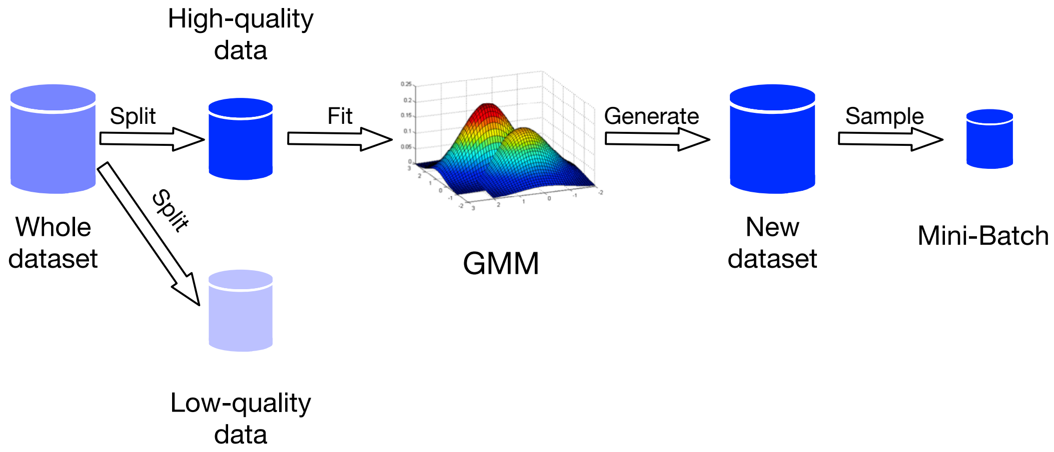
- Participating agents
  - Equally share the overall reward
- Non-participating agents:
  - 0 reward



How to improve training data quality in experience replay: GMM based rectified sampling



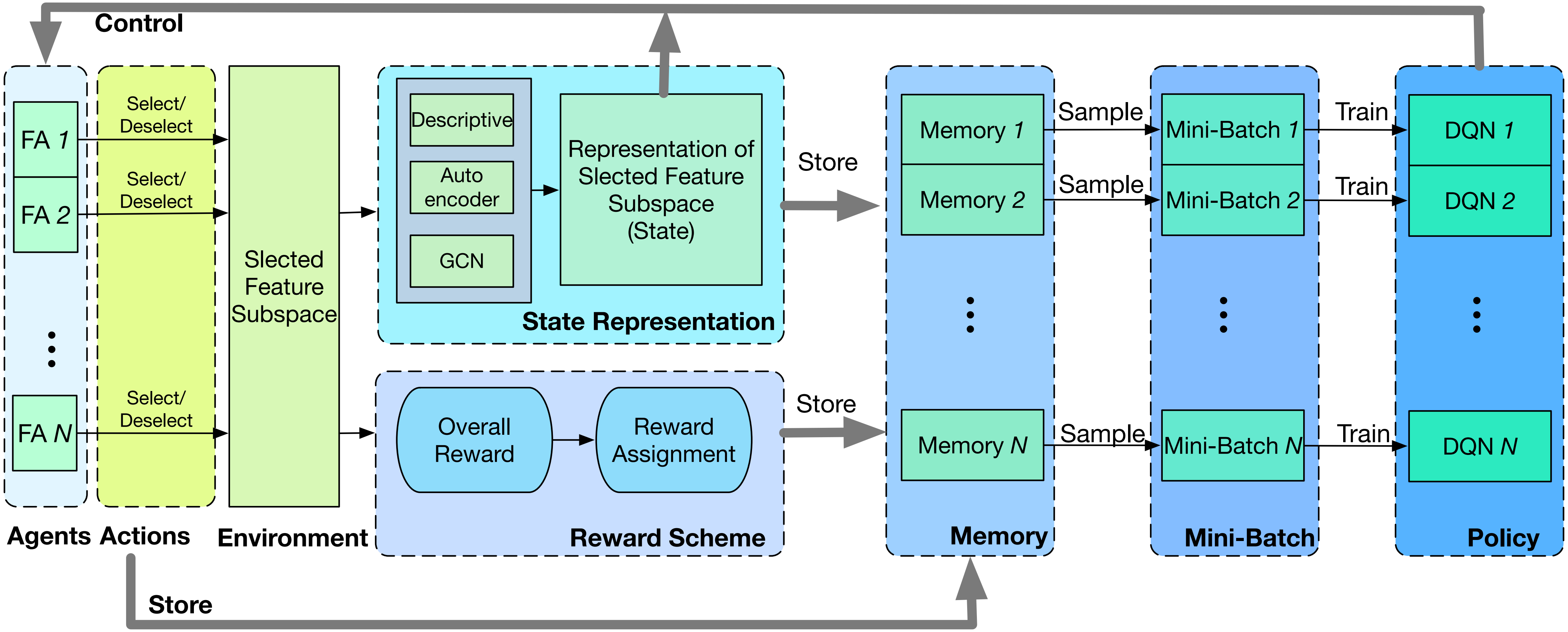
Conventional sampling strategy.



GMM based sampling strategy.

- Modeling heterogeneity of data samples via mixture model based rectified sampling
- Promoting diversity and coverage of sampling strategies

Recap: the framework of RFSL

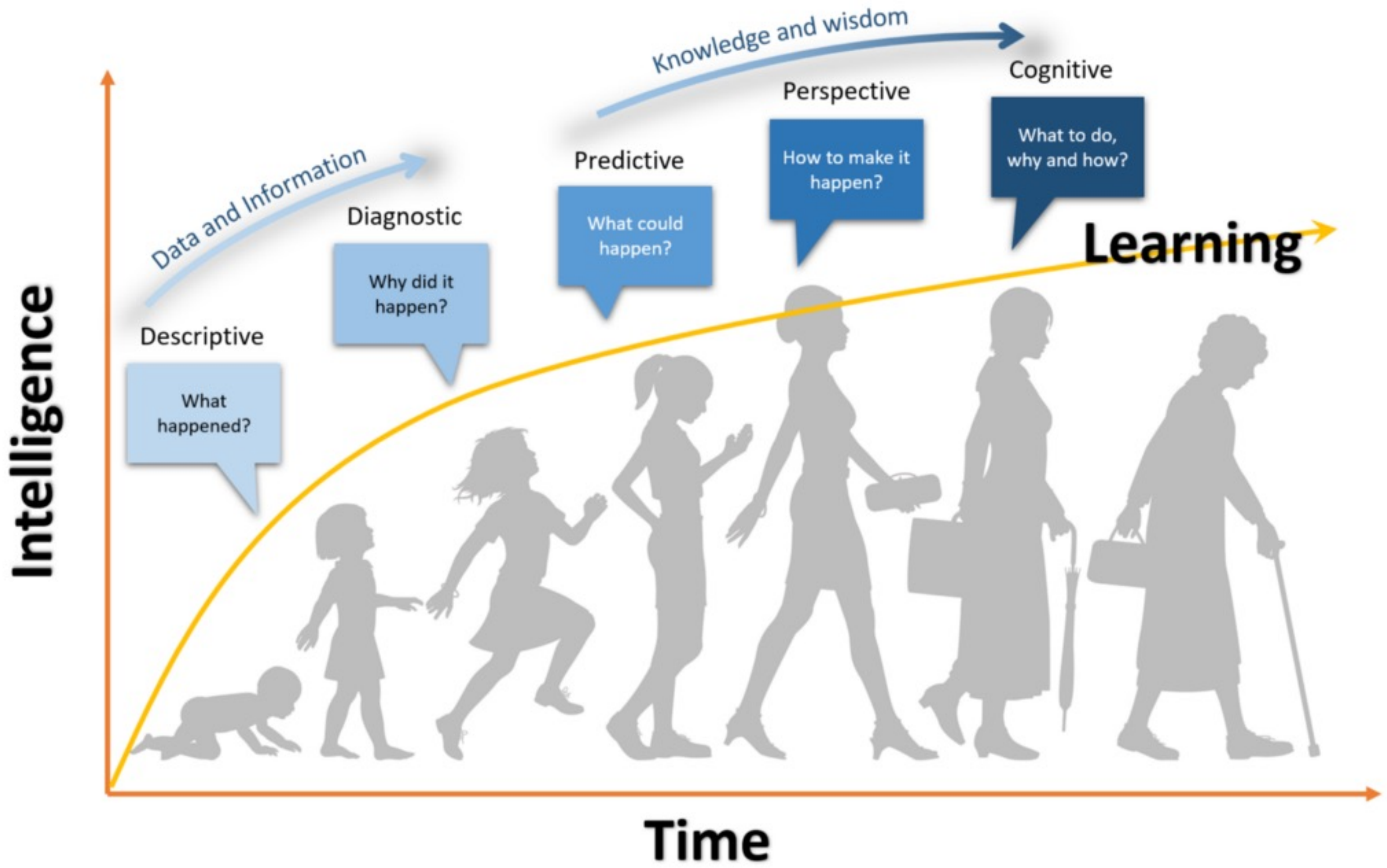


Can our study improve feature selection performance?

		Predictors				
		RF	LASSO	DT	SVM	XGBoost
Algorithms	K-Best	0.7943	0.8246	0.8125	0.8324	0.8076
	mRMR	0.8042	0.8124	0.8096	0.8175	0.8239
	LASSO	0.8426	**0.8513**	0.8241	0.8131	0.8434
	RFE	0.8213	0.8236	0.8453	0.8257	0.8348
	GFS	0.8423	0.8318	0.8350	0.8346	0.8302
	SARLFS	0.8321	0.8295	0.8401	0.8427	0.8450
	MARLFS	**0.8690**	0.8424	**0.8583**	**0.8542**	**0.8731**

- Benchmark application
  - Data: 15120*54, 7 labels; Task: Classification
- Baselines:
  - K-Best Selection, mRMR, LASSO, Recursive Feature Elimination (RFE), Genetic Feature Selection (GFS), Single-Agent Reinforcement Learning Feature Selection (SARLFS)
- Evaluation Metrics: overall classification accuracy

Human learning



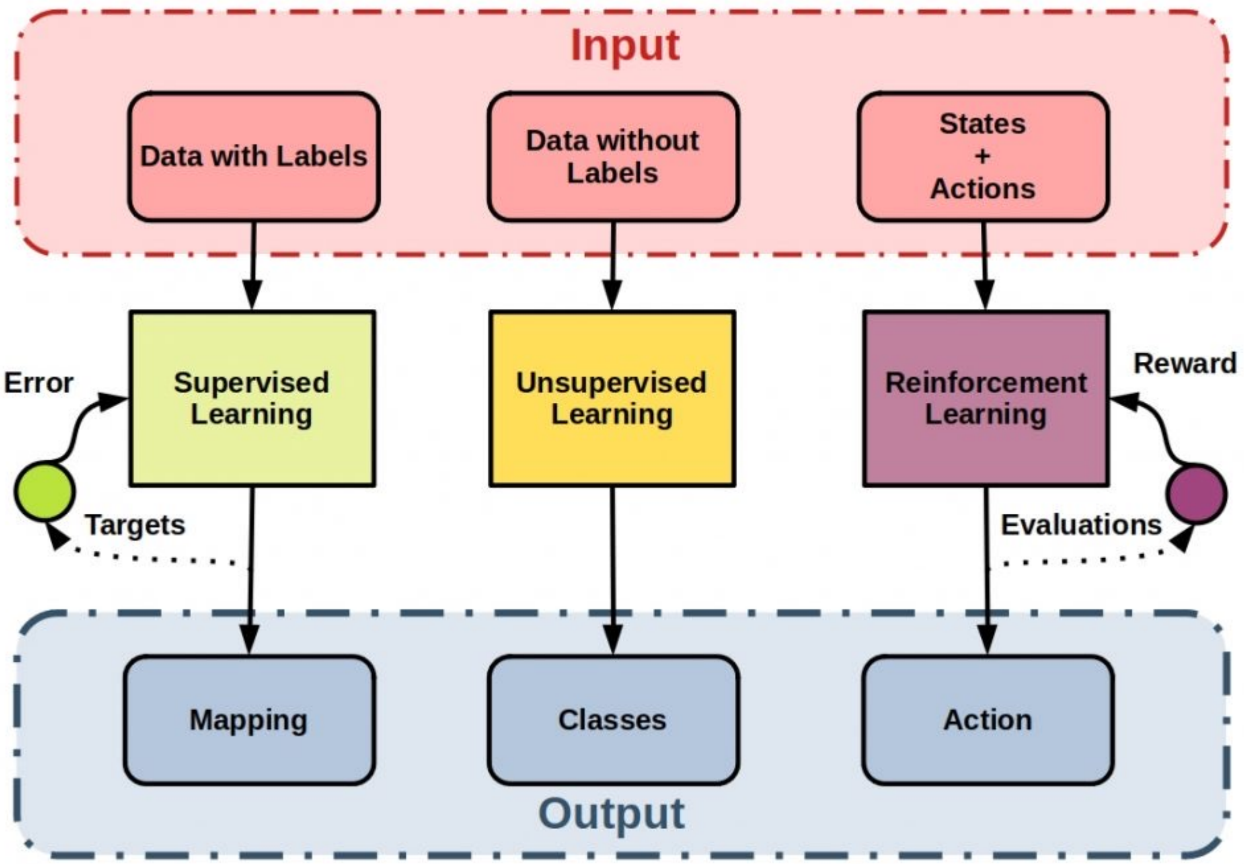
Humans learn vertically and horizontally

- Learning ``vertically''
  - Supervised learning of historical successes and mistakes
- **Gibson's theory of development (Eleanor Gibson)**
- Learning ``horizontally''
  - Interactive learning from peer experiences in the same problem domain

"The more chances they are given to perceive and interact with their environment, the more affordances they discover, and the more accurate their perceptions become."

Eleanor Gibson

Machines are limited in interactive abilities



Issue 1: Supervised and unsupervised machines have limited interaction abilities

Issue 2: Reinforcement learning (RL) interacts with task evaluators in the environment, but limited interaction with peer experiences

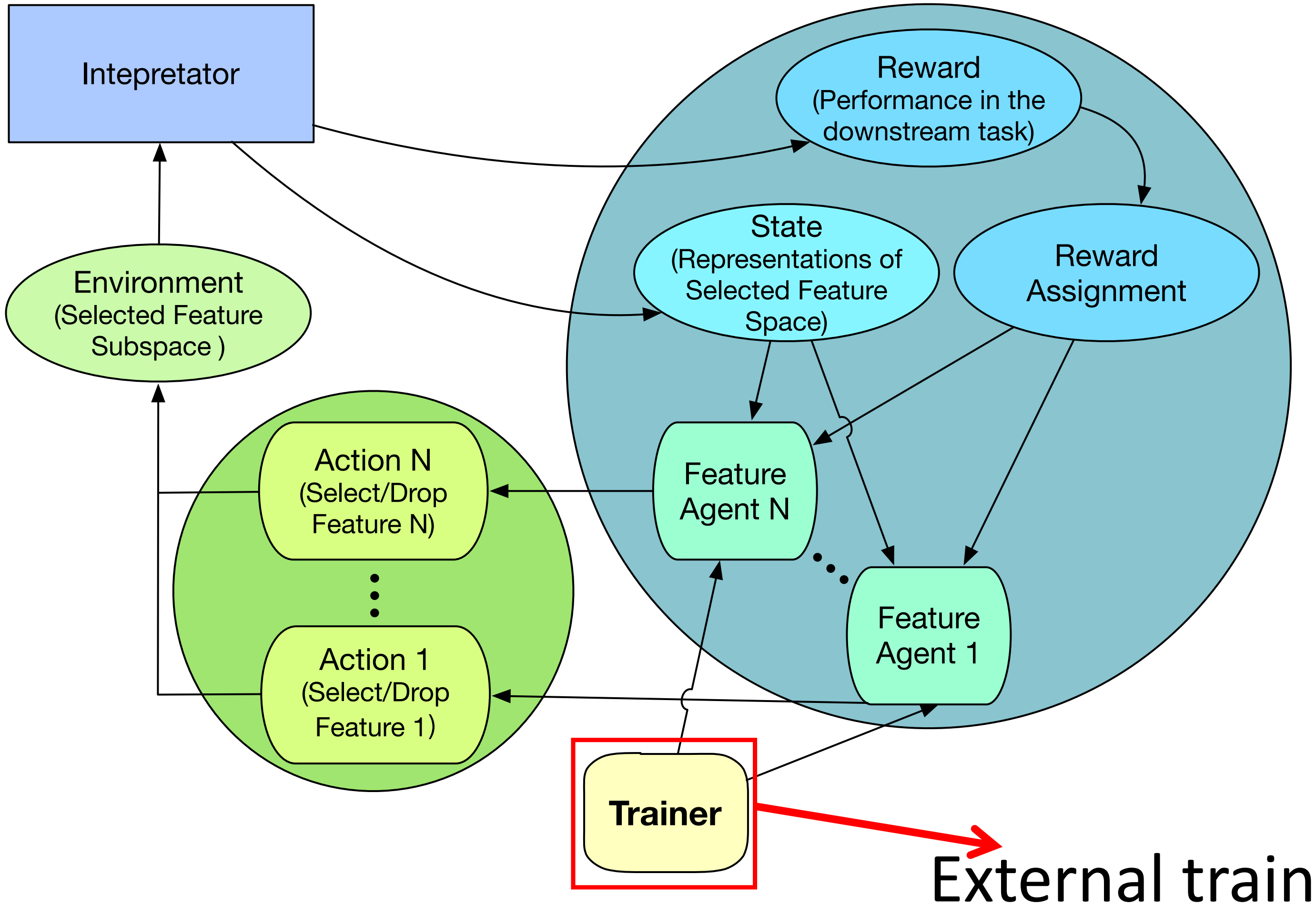
A finding: interactive learning as supervision signal to robustize reinforcement

- Reinforcement learning: generating data while learning via trials and errors
  - Strength: self-optimizing, doesn't need training data
  - Weakness: hard to tune and slowly grow quality policies
- Supervised learning
  - Strength: reliable success rate
  - Weakness: need lots of training data
- *Can we integrate supervision with reinforcement?*
  - *Let external trainers and prior knowledge in the same task domain interact with reinforcement agents to guide agent learning*

Interactive Reinforcement

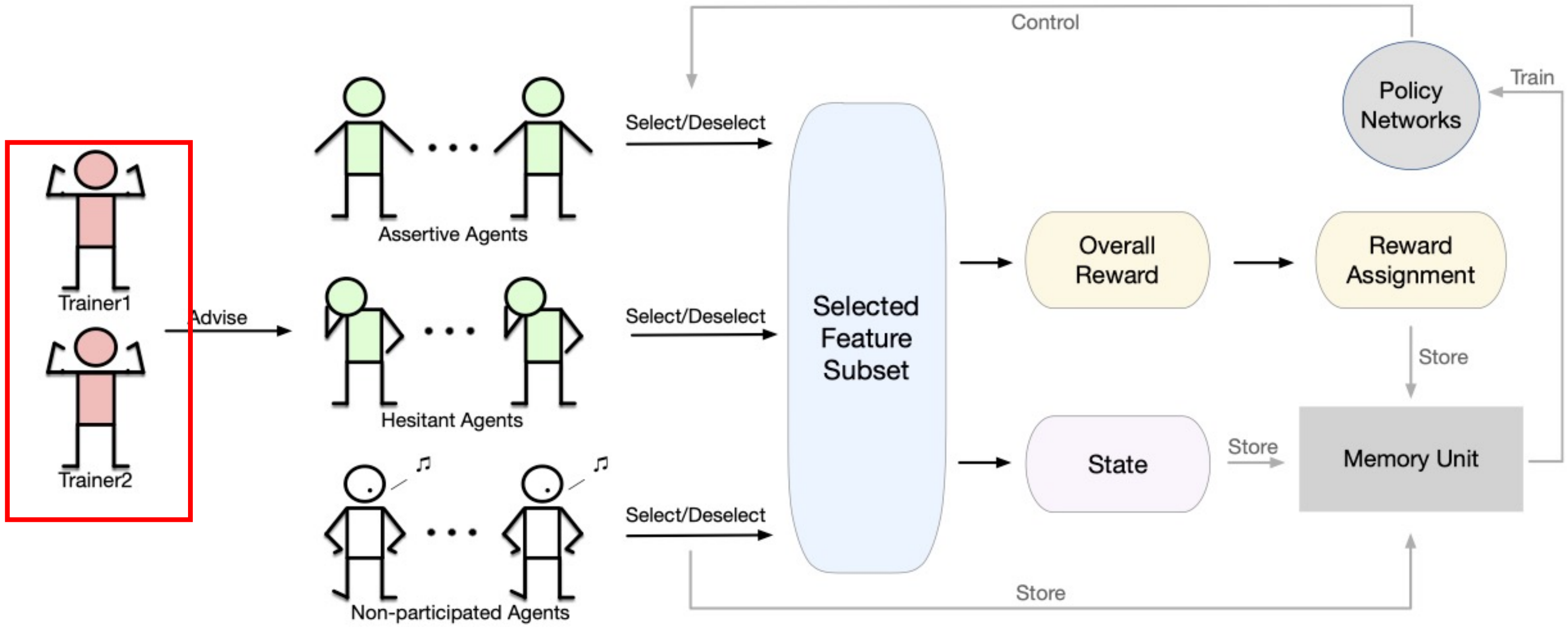
Peer Experiences as Supervision + Adapt Past and Peer Experiences into Knowledge

Overview of interactive RFSL for representation subspace identification



External trainer advising as supervision

Diversity-aware interaction mechanism (1): diversified external trainers



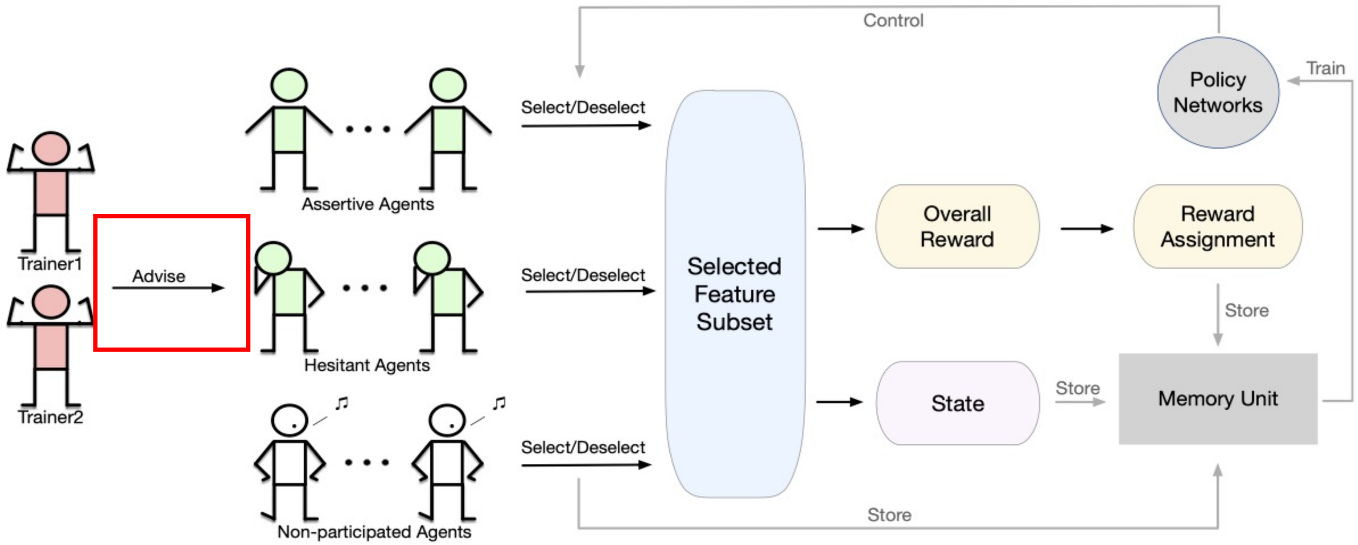
We propose multiple external trainers:

✓ **KBest based trainer**

✓ **Decision Tree based trainer**

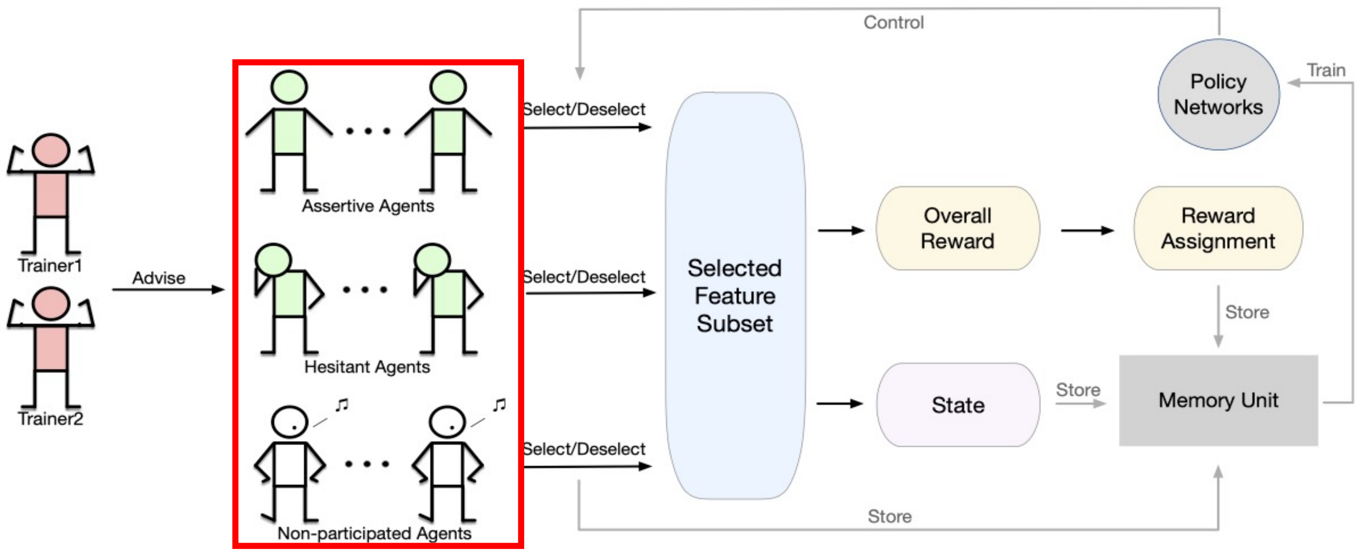
✓ **mRMR**

Diversity-aware interaction mechanism (2): diversified participated features



Diversify the selection of various features as the input of trainers (traditional feature selection methods) to generate diverse advice

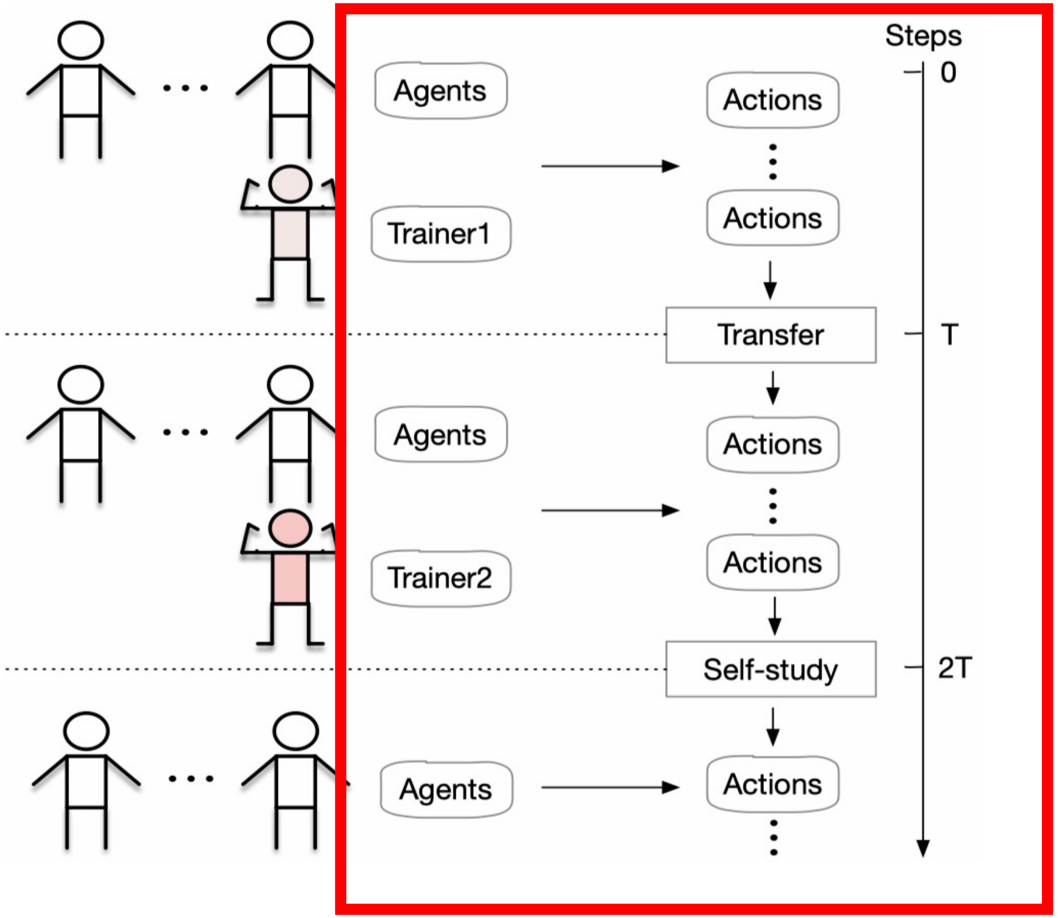
Diversity-aware interaction mechanism (3): diversified advice for different agents



- Assertive agents: do not need to follow advices from external trainers
- Hesitant agents: follow advices from external trainers

Diversity-aware interaction mechanism (4): diversified hybrid teaching strategy

- Different trainers provide advice at different stages

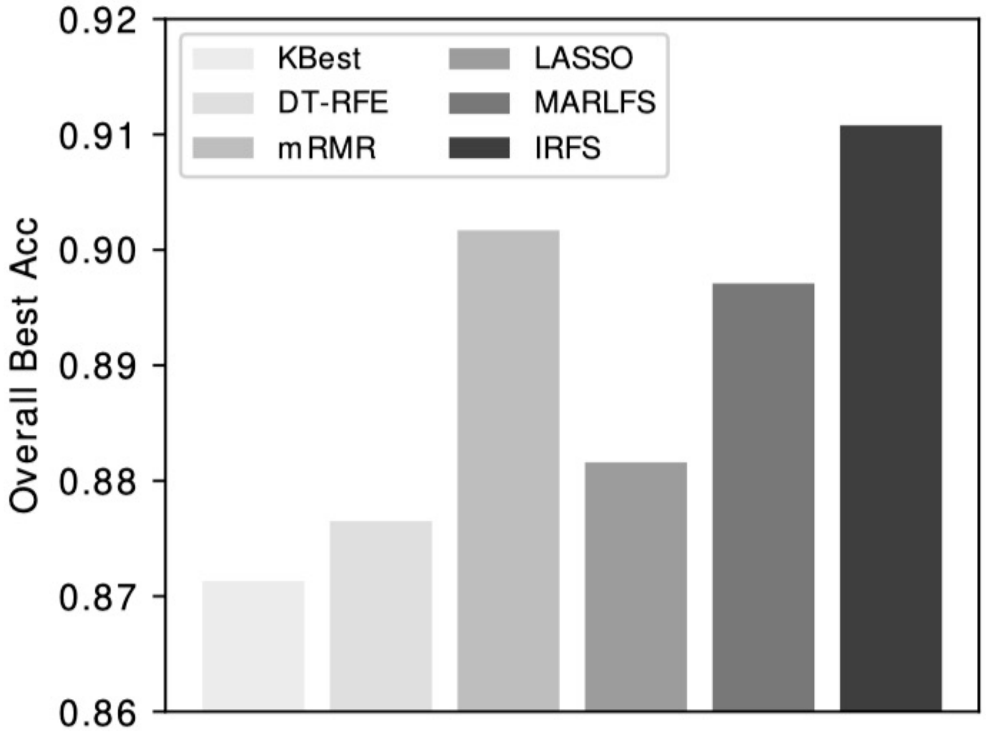


Evaluation Dataset

- Public ally available Kaggle competition datasets
  - Forest Cover (FC), Spambase, Insurance Company Benchmark (ICB), MUSK

	ForestCover	Spambase	ICB	Musk
Features	54	57	86	168
Samples	15120	4601	5000	6598

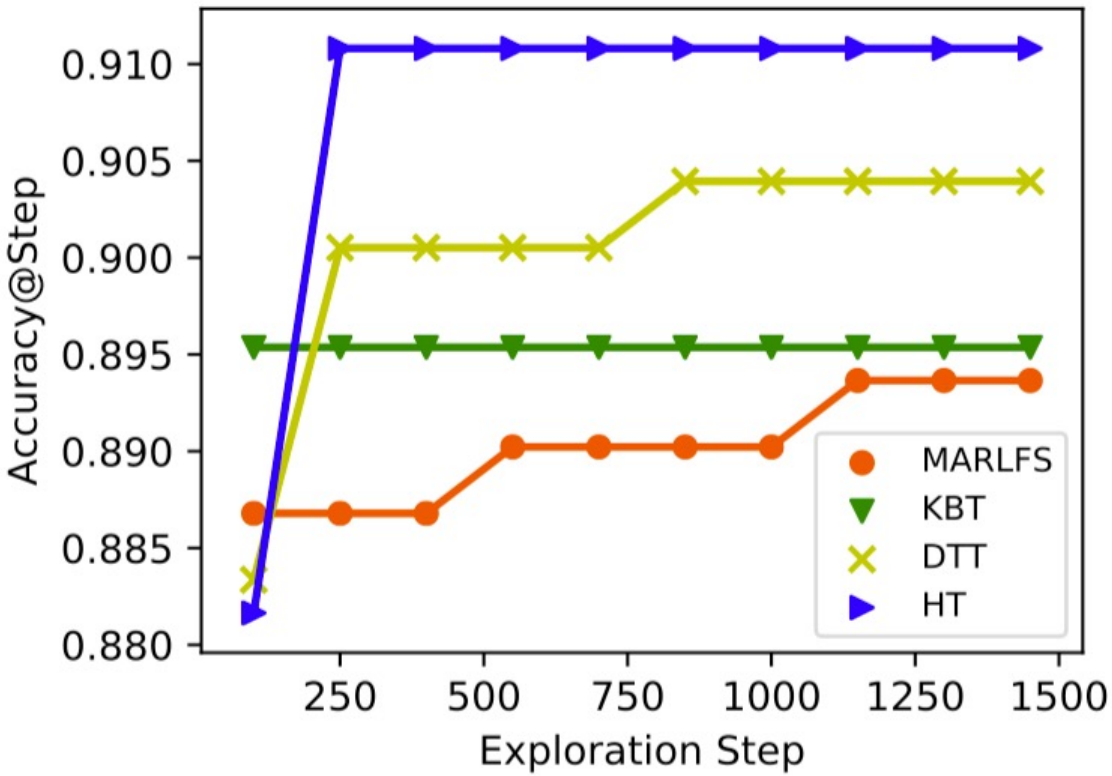
Can our method improve feature selection performance? Results on ICB



- Baselines:
  - **KBest** Feature Selection
  - **mRMR**
  - **DT-RFE** (Decision Tree Recursive Feature Elimination)
  - **LASSO**
  - **MARLFS** (Multi-agent Reinforcement Learning Feature Selection)

Our method: **IRLFS**.
For the accuracies,
the **higher**, the **better**.

Can our interactive strategy improve learning efficiency? Results on ICB



- Variants
  - MARLFS: without any trainer
  - KBT: IRLFS with KBest based trainer
  - DTT: IRLFS with decision tree based trainer
  - HT: IRLFS with Hybrid Teaching by KBT and DTT
- Metrics:
  - Best Accuracy@Step

Explainable and Optimal Representation Space Reconstruction: A Generation Perspective

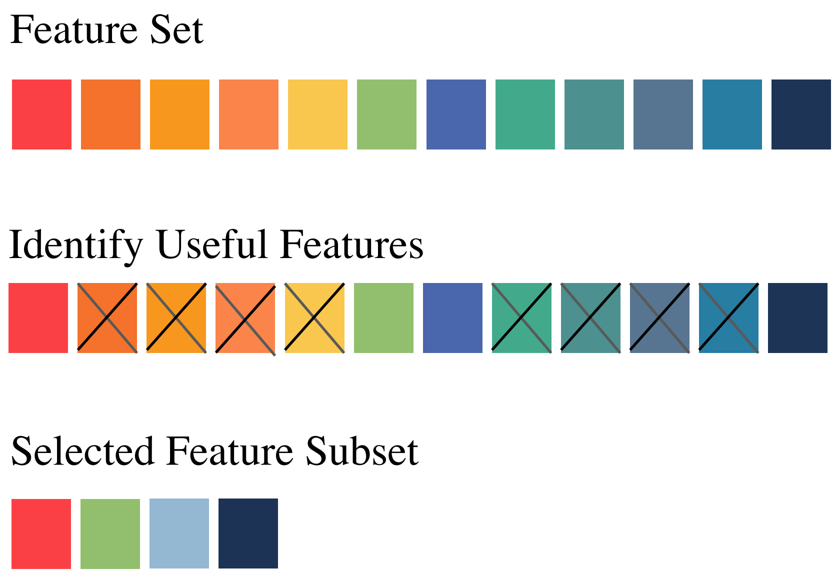
DNNs create new features in a latent space

- DNNs create new features at multi-levels of abstraction
  - Capture variation patterns in an embedding space
  - Transform the data into a set of principal components
  - Remove redundancy in representation
  - Handle indirect relationship between features and goals

- When DNNs outperform linear regression on a selected feature subset
  - Reason: the selected features are not complete and optimal, while DNNs create newer, more complete, discriminative dimensions
  - Can we imitate the feature generation capability of DNNs in an explicit space?

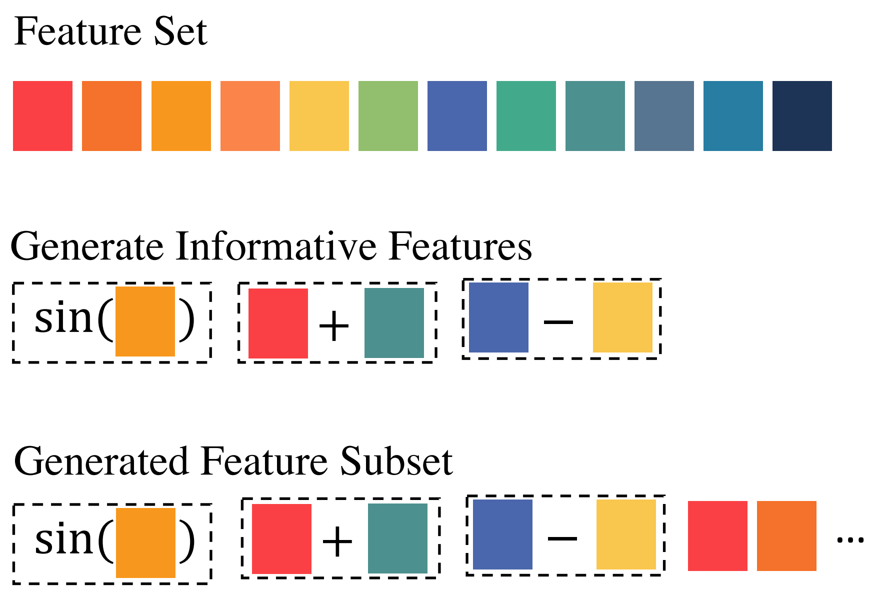
Can machines imitate DNNs to create new, features in an explicit space?

- Feature Selection



Philosophy: Improve the representation space from a reduction perspective

- Feature Generation

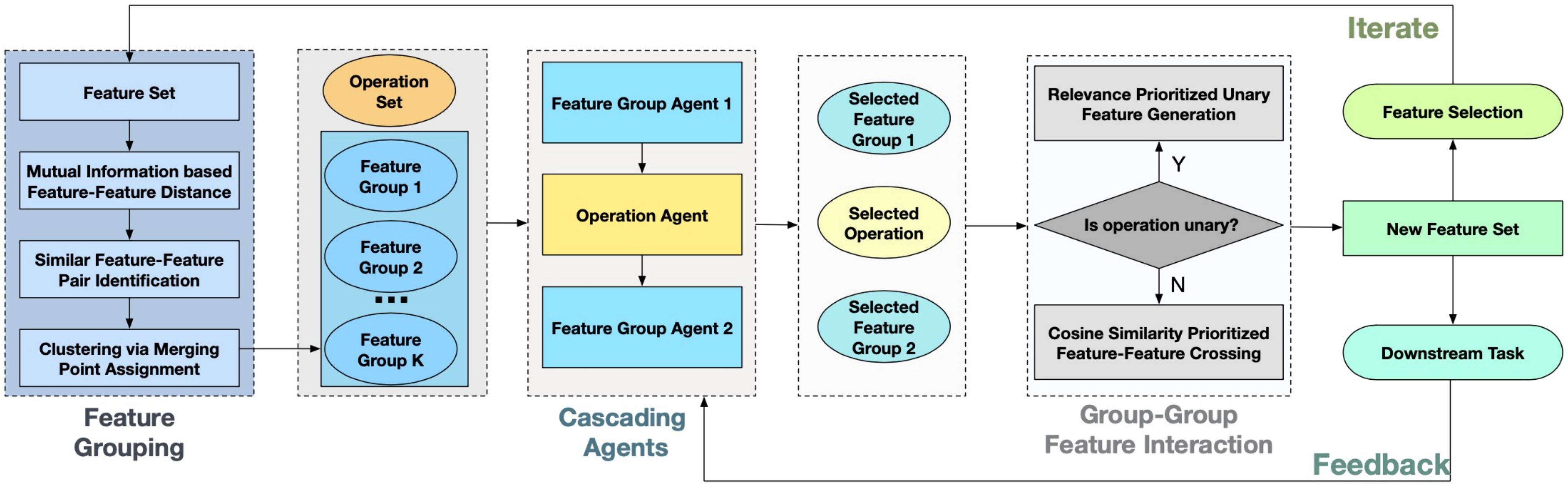


Philosophy: Improve the representation space from an addition perspective

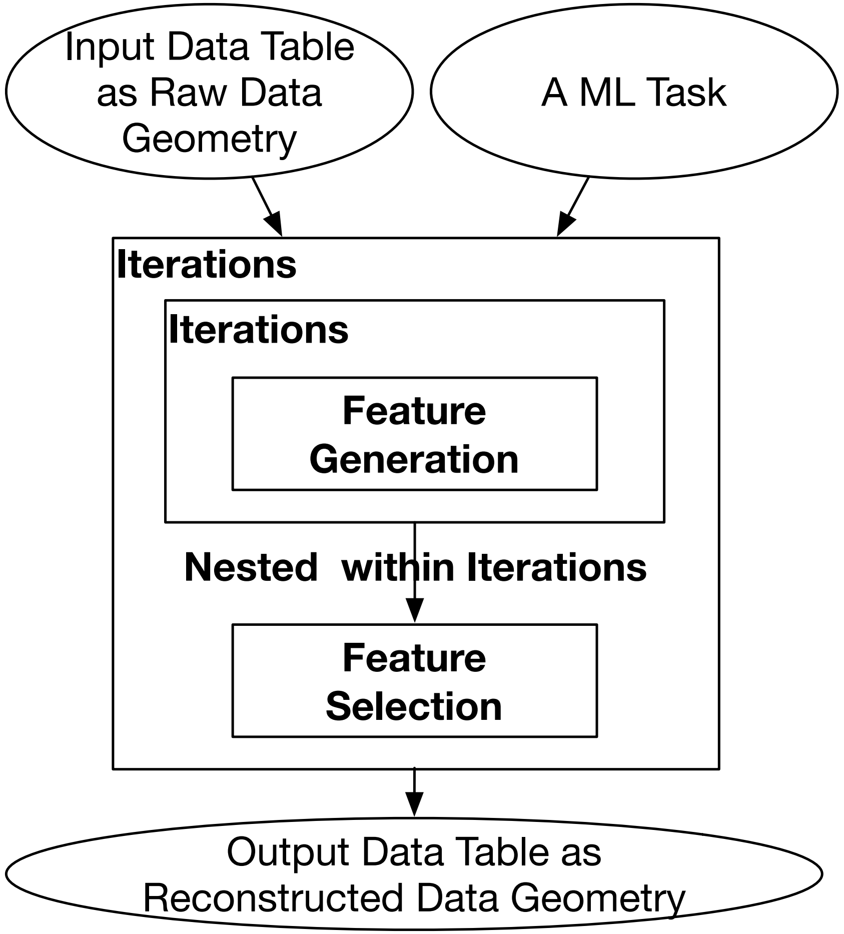
Proposed Solution: Group-wise reinforcement feature generation learning

Reinforcement feature generation learning (RFGL): learn a feature generator that

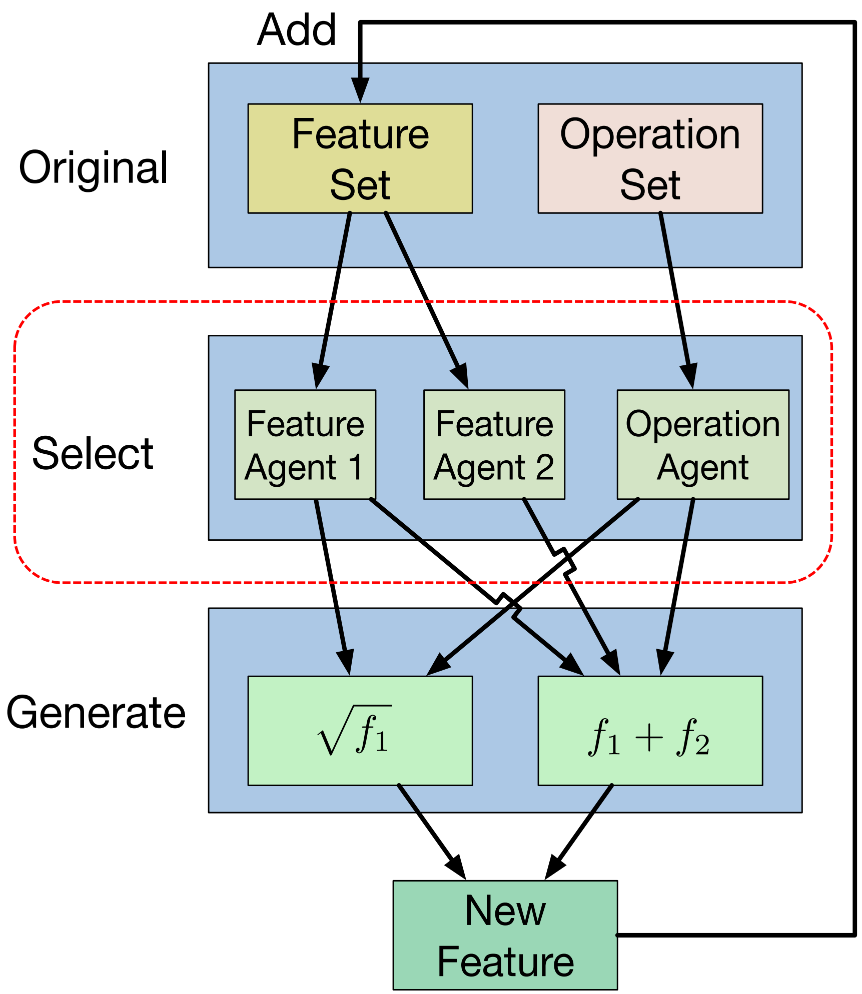
1. Explainable Explicitness: explicit generation process with semantic labels
2. Self-optimizing: automatically reconstruct a new feature space
3. Efficiency and reward augmentation: group-wise generation



Goal 1: a principled iterative nested generation and selection framework: elements and structure

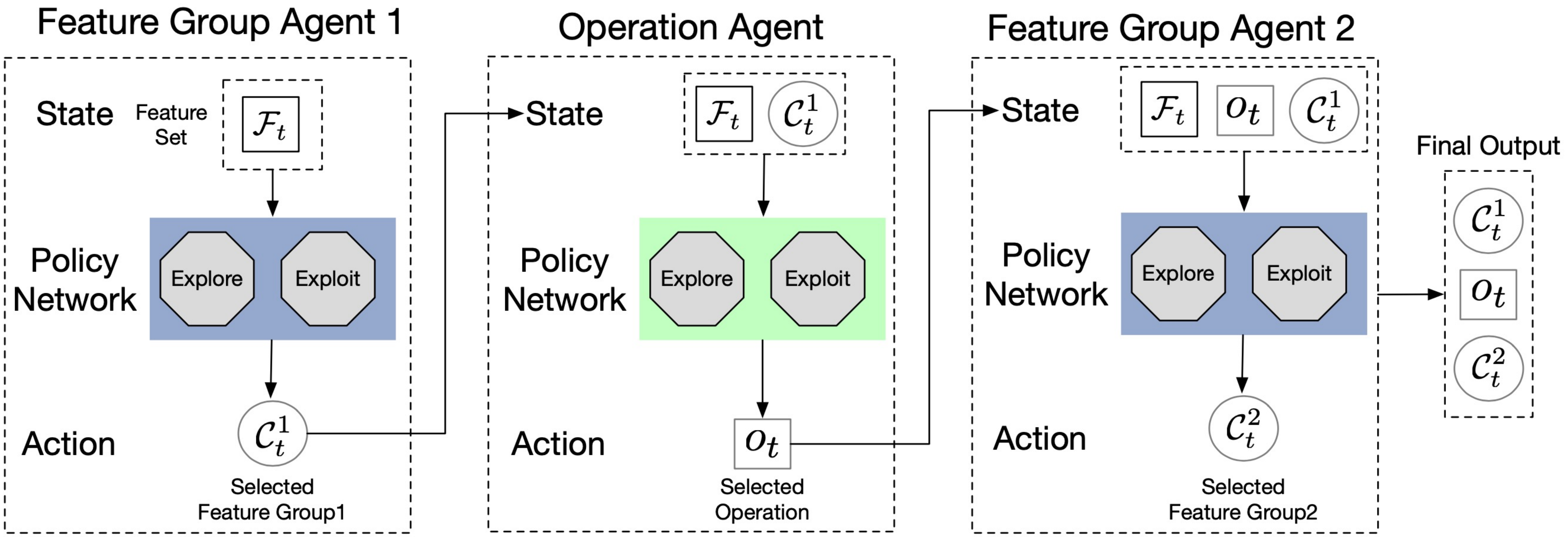


Goal 2: cascading reinforcement learning for automated and explicit feature generation

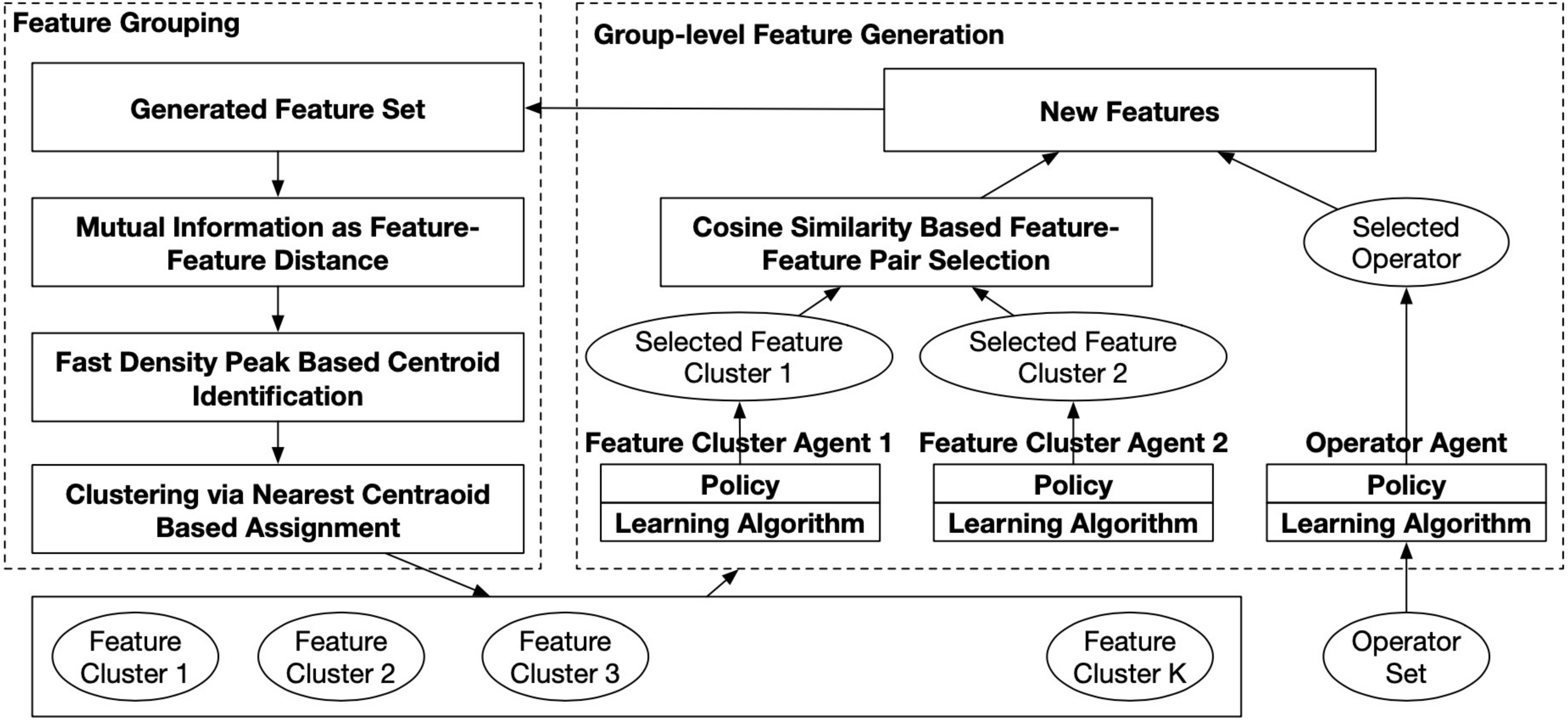


- Feature Agent 1---> Feature 1
- Operation Agent ---> Operation
- If operation is unary:
  - Conduct Operation on Feature 1
- If operation is binary:
  - Feature Agent 2---> Feature 2
  - Conduct Operation on Feature 1 and Feature 2

Cascading state sharing



Goal 3: group-wise feature generation



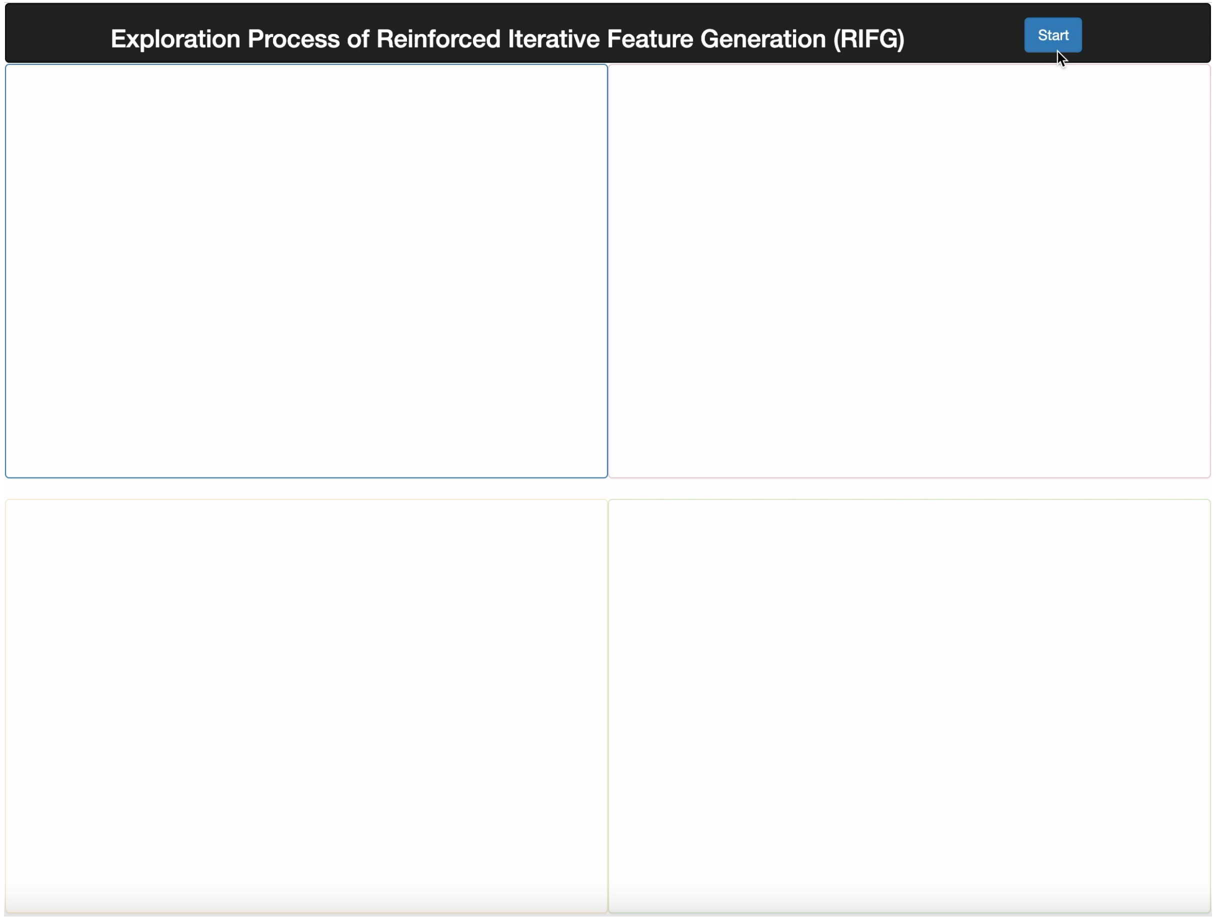
Demo setup: datasets, tasks, and algorithms

- Feature set:
  - f_1:Frequency
  - f_2: Angle of attack
  - f_3: Chord length
  - f_4: Free-stream velocity
  - f_5: Suction side displacement thickness
- Task:
  - To predict if the scaled sound pressure level is larger than threshold (**classification**)
- Algorithm:
  - Random forest classifier
- Platform:
  - CPU: I9-9920X 3.50GHz, memory: 128GB memory, operation system: Ubuntu 18.04 LTS

□ **Operation Set :**

{sqrt, square, sin, cos, tanh, +, -, *}

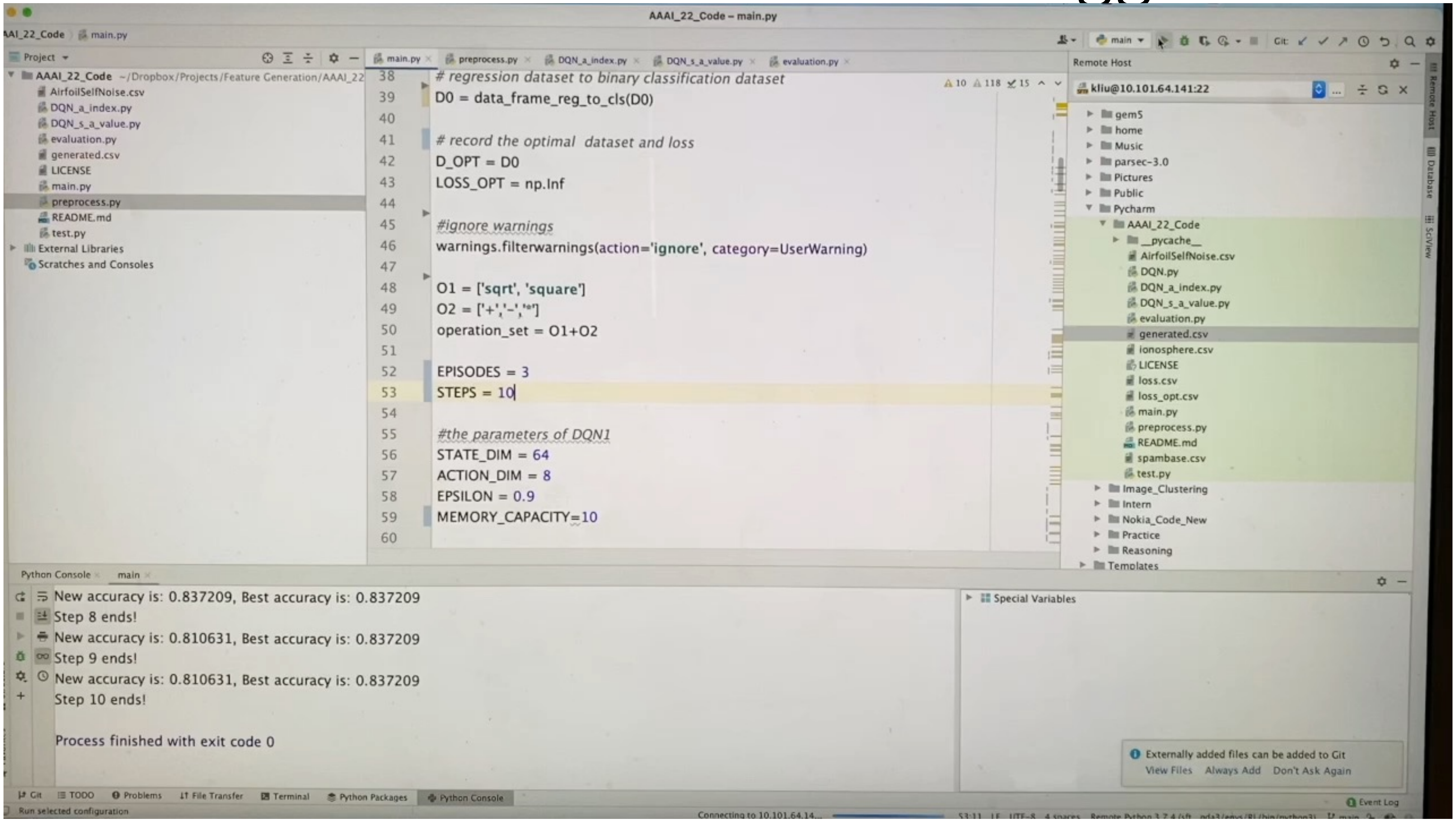
Algorithmic tool and demonstration systems



Is RIFG fast and interpretable?

- Time elapsed:
  - 7 mins 37 secs
- Improvement:
  - Accuracy: 0.824 ->0.907 10.07%
  - Precision: 0.822 ->0.909 10.58%
  - Recall: 0.863->0.916 6.14%
  - F1: 0.842 ->0.909 7.95%
- Generated features (best accuracy):
  - $\{f_1, tanh(f_1), sin(f_3 * f_5), (f_2 - f_1 * f_5)^2, \sqrt{f_4}, cos(tanh(f_1)), cos(sin(f_2 - (f_2 * f_4)^2)), (f_2 * f_4 * f_5 + {f_1}^2 * f_4)^3\}$

Demo: take 25 seconds to improve Recall from 0.816 to 0.879 as A Bot in Kaggle



Conclusion Remarks

- Optimizing data representation space versus optimizing model structure space?
  - From models to data
- Latent representation learning versus explicit representation reconstruction
  - From empirical and handcrafted to automated
  - From latent to explicit
  - From Blackbox to explainable and traceable
- A wide range of applications
  - Better prediction: Your current deployed ML systems (e.g., recsys) can do a better job
  - Green computing: You can use a simple model with the optimized data reconstructed by our tool to achieve similar performances with complex deep models of large parameters
  - Representation as a tool of user, system, product, location profiling and characterization
- How reliable is our method for practical deployment?
  - Automated, explainable, traceable, take some time to explore but time costs can be reduced by pre-training via offline RL with data in the same problem domain
  - Convert agents' decisions into a task of generating decision sequences and optimize the performances in a continuous space via embedding