

# REvolve: Reward Evolution with Large Language Models using Human Feedback



RISHI HAZRA\*



ALKIS SYGKOUNAS\*



ANDREAS PERSSON



AMY LOUTFI



PEDRO ZUIDBERG  
DOS MARTIRES

\* equal contribution

# REvolve: Reward Evolution

2

- *We know more than we can tell* (Polanyi's Paradox)

- **What** is REvolve?
  - LLM-based Reward Design Framework for Reinforcement Learning
- **Why** do we need it?
  - Reward Design is hard (think Autonomous Driving).
  - "Good behavior" is subjective.
  - REvolve automates *reward generation* and refines them with *behavior verification* by humans.

```
def reward_fn(speed, collision, min_pos, distance):  
    reward_components = {}  
  
    # Speed reward component  
    speed_reward = np.exp(-speed_temp * speed_difference)  
    reward_components['speed_reward'] = speed_reward  
  
    # Collision reward component (high penalty for collision)  
    collision_reward = -100 if collision else 0  
    reward_components['collision_reward'] = collision_reward  
  
    # encouraging safe distance maintenance  
    distance_reward = np.exp(-distance_temp * max(0, 20 - distance))  
    reward_components['distance_reward'] = distance_reward  
  
    ...  
  
    # Calculate total reward  
    total_reward = sum(reward_components.values())  
    total_reward = np.clip(total_reward, -1, 1)  
  
    return total_reward
```

# REvolve Framework

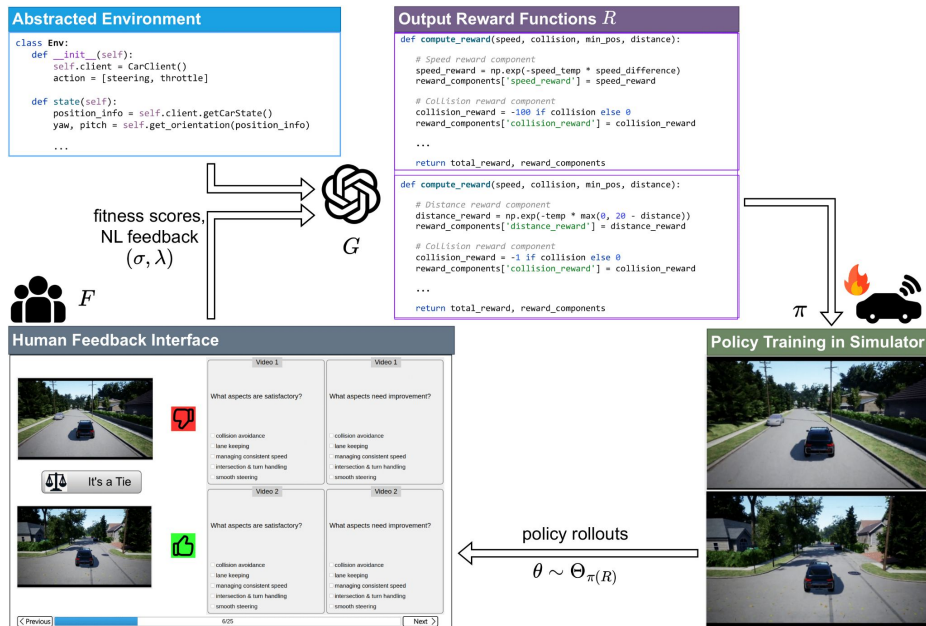
3

REvolve: Reward Evolution with Large Language Models using Human Feedback, Hazra and Sygkounas et al, ICLR 2025

## Contributions:

- Evolutionary Algorithms for Reward Design
- Human as fitness functions.
- Elo rating system for ranking

*minimal human involvement*



# Related Works

4

REvolve: Reward Evolution with Large Language Models using Human Feedback, Hazra and Sygkounas et al, ICLR 2025

	<b>Expert</b>	<b>Eureka [1]</b>	<b>REvolve</b>
Reward Generation	Manual	Automated	Automated
Fitness Function	Not required	Required	Not required
Search Strategy	Trial & Error	Greedy Iterative	Evolutionary
Feedback Source	-	Automated	Human / Automated

[1] Eureka: Human-Level Reward Design via Coding Large Language Models, Ma et al, ICLR 2024

# LLM Operators

5

REvolve: Reward Evolution with Large Language Models using Human Feedback, Hazra and Sygkounas et al, ICLR 2025

```
def compute_reward(speed, collision, min_pos, distance):  
    # Initialize dictionary to hold reward components  
    reward_components = {}  
  
    # Calculate the speed reward component  
    speed_reward = np.exp(-speed_temp * speed_difference)  
    reward_components['speed_reward'] = speed_reward  
  
    # Calculate the collision reward component  
    collision_reward = -1 if collision else 0  
    reward_components['collision_reward'] = collision_reward  
  
    # Ensure enough distance in front of the car to avoid collisions  
    distance_reward = np.exp(-distance_temp * max(0, 20 - distance))  
    reward_components['distance_reward'] = distance_reward  
  
    smoothness_reward = np.exp(-temp * abs(angular_velocity_z))  
    smoothness_reward = np.exp(-temp * np.mean(steering_diffs))  
  
    reward = sum(reward_components.values())  
    reward = np.clip(reward, -1, 1)  
  
    return total_reward, reward_components
```

Mutation

```
def compute_reward(speed, collision, min_pos, distance):  
    reward_components = {}  
  
    # Encourage the agent to maintain speed between 9.0, 10.5 m/s.  
    speed_reward = -np.abs(speed - 9.75) # 9.75 is the midpoint  
    speed_reward = np.clip(speed_reward, -1, 0)  
    speed_reward = np.exp(speed_reward / speed_temp)  
    reward_components['speed_reward'] = speed_reward  
  
    # heavily penalize the agent if a collision occurs.  
    collision_reward = -100 if collision else 0  
    reward_components['collision_reward'] = collision_reward  
  
def compute_reward(speed, collision, min_pos, distance):  
    # Initialize dictionary to hold reward components  
    reward_components = {}  
  
    # Calculate the speed reward component  
    speed_reward = np.exp(-speed_temp * speed_difference)  
    reward_components['speed_reward'] = speed_reward  
  
    # Calculate the collision reward component  
    collision_reward = -1 if collision else 0  
    reward_components['collision_reward'] = collision_reward  
  
    # Ensure enough distance in front of the car to avoid collisions  
    distance_reward = np.exp(-distance_temp * max(0, 20 - distance))  
    reward_components['distance_reward'] = distance_reward  
  
    ...  
  
    reward = sum(reward_components.values())  
    reward = np.clip(reward, -1, 1)  
  
    return total_reward, reward_components
```

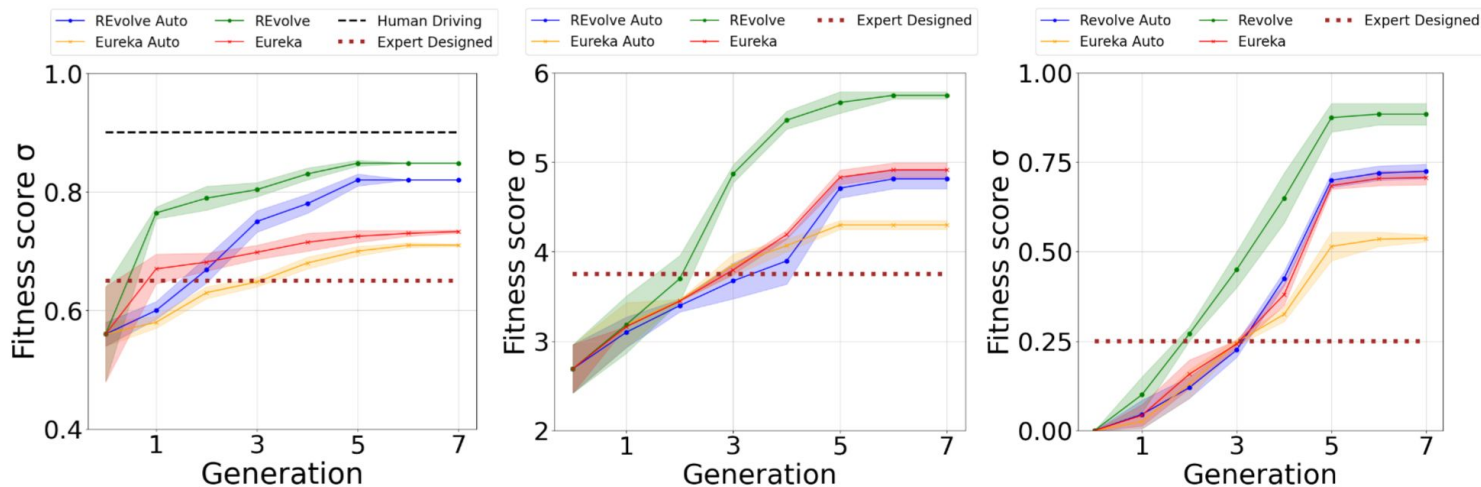
Crossover

```
def compute_combined_reward(speed, collision, min_pos, distance):  
    reward_components = {}  
  
    # Speed reward component  
    speed_reward = np.exp(-speed_temp * speed_difference)  
    reward_components['speed_reward'] = speed_reward  
  
    # Collision reward component (high penalty for collision)  
    collision_reward = -100 if collision else 0  
    reward_components['collision_reward'] = collision_reward  
  
    # encouraging safe distance maintenance  
    distance_reward = np.exp(-distance_temp * max(0, 20 - distance))  
    reward_components['distance_reward'] = distance_reward  
  
    ...  
  
    # Calculate total reward  
    total_reward = sum(reward_components.values())  
    total_reward = np.clip(total_reward, -1, 1)  
  
    return total_reward, reward_components
```

# Results: Designed Fitness Function

6

REvolve: Reward Evolution with Large Language Models using Human Feedback, Hazra and Sygkounas et al, ICLR 2025

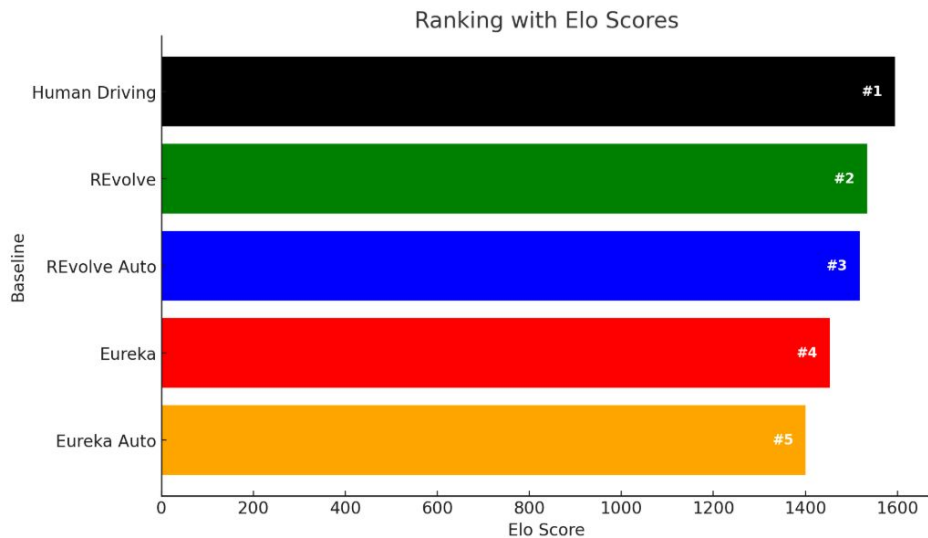


REvolve > REvolve Auto > Eureka > Expert Designed

# Results: Human Evaluation

7

REvolve: Reward Evolution with Large Language Models using Human Feedback, Hazra and Sygkounas et al, ICLR 2025



REvolve > REvolve Auto > Eureka > Expert Designed

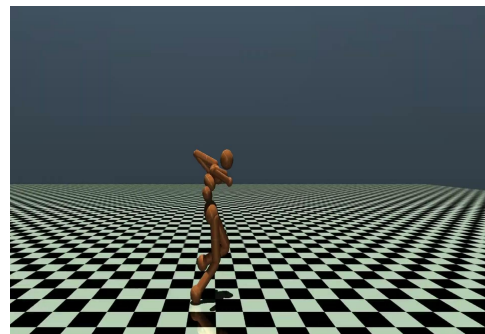
# REvolve vs. Eureka

8

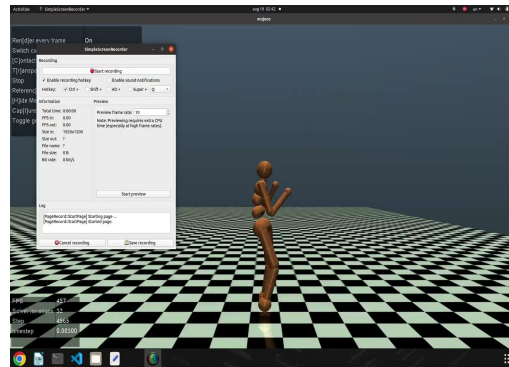
REvolve: Reward Evolution with Large Language Models using Human Feedback, Hazra and Sygkounas et al, ICLR 2025

Eureka: Human-Level Reward Design via Coding Large Language Models, Ma et al, ICLR 2024

REvolve



Eureka





# REvolve: Reward Evolution with Large Language Models using Human Feedback

9



**Saturday, 26th (9:00-11:30)**

<https://rishihazra.github.io/REvolve/>