Absolutely! Here's the full English translation of your message and report, preserving the bilingual technical terms and formatting:

👍 Sure! I'll combine **Part 1 + Part 2** into a complete report written in Chinese, but I'll provide both Chinese and English for **keywords and technical terms**, such as "強化學習 (Reinforcement Learning)." This way, your readers can understand more easily while maintaining technical accuracy.

EVAT Reinforcement Learning Report (EVAT 強化學習實驗報告)

Part 1: Background (背景)

Dataset Preparation (資料集準備)

To train a Reinforcement Learning (RL, 強化學習) model that simulates user interactions in electric vehicle application scenarios, we need to prepare a simulated dataset. Since we currently lack real user behavior data, we designed a **Random Action Generator (隨機行為生成器)** to mimic user interactions within the app.

These actions correspond to six types in the gamification system:

- Check-in (打卡)
- Report fault (故障回報)
- Validate AI prediction (AI 驗證)
- Discover new station in black spot (黑點發現)
- Use route planner (路線規劃)
- Ask chatbot question (聊天機器人提問)

We assigned different probabilities to each action—common behaviors like check-in and chatbot questions appear more frequently, while rare but important actions like discovering black spots occur less often. Additionally, we included a **10% chance of “no action” (不做任何動作)** to simulate users who may not interact at all.

The final dataset is saved as `simulated_user_actions.json`, serving as the foundation for environment training.

Environment Design (EVATGamificationEnv 環境設計)

We designed a custom environment `EVATGamificationEnv` following the OpenAI Gym/Gymnasium format.

- **Action Space (動作空間)**
  Six discrete actions corresponding to the six interaction types mentioned above.
- **Observation Space (觀察空間)**
  A vector that records the number of times each action is selected, normalized to values between 0 and 1.
- **Reward Shaping (獎勵塑形)**
  Each action has a base reward—for example, check-in is +10 points, while discovering a black spot is +120 points. We added three reward adjustments:
  i. **Reward Scaling & Clipping (報酬縮放與裁切)**: All rewards are divided by 100 and clipped to the range [-1, 1].
  ii. **Diminishing Returns (遞減回報)**: Repeated use of the same action gradually reduces its reward.
  iii. **Diversity Bonus (多樣性獎勵)**: If two consecutive actions are different, an extra +2 points is awarded.

These designs encourage the model to **explore diverse behaviors**, rather than repeatedly choosing a single high-reward action.

Training Purpose (訓練目標)

The goal of this experiment is to use reinforcement learning algorithms—specifically **PPO (Proximal Policy Optimization)**—to help the agent learn an optimal behavior strategy that maximizes total reward within a limited number of interaction steps.

This not only demonstrates the concept of gamification system design but also lays the groundwork for optimizing future real-user data.

Part 2: Training & Evaluation (訓練與評估)

Training Setup (訓練設定)

We used the **PPO algorithm** from the `stable-baselines3` library, and built a **Vectorized Environment (向量化環境)** and **Monitor (監控器)** to record the training process.

Core code:

```
# [Code remains unchanged]
```

Evaluation Results (評估結果)

During training, we evaluated every **200 timesteps**, recording average rewards and episode lengths.

- Initial average reward: **13–15**
- Final stabilized average reward: **around 24**
- Episode length increased from **30–40 steps** to nearly **60 steps**, indicating the agent learned to extend interactions rather than ending episodes prematurely.

We plotted a **Smoothed Evaluation Reward Curve (平滑化的評估獎勵曲線)** showing a clear rise from low values to a stable plateau, indicating that the PPO strategy has converged to a stable behavior.

Behavior Analysis (行為分析)

To understand the agent's strategy, we recorded the action distribution from a post-training episode:

- **Action Distribution (動作分佈)**: The model favors high-value actions (e.g., discovering black spots, AI validation), but due to the diversity bonus, it occasionally tries other actions.
- **Average Reward per Action (動作平均回報)**: Clear differences in reward across actions explain the model's preference for certain high-reward behaviors.

Interpretation (結果解讀)

1. **Convergence (收斂效果)**: The model quickly found a high-reward strategy and maintained it, aligning with the reward shaping design.
2. **Preference Bias (偏好傾向)**: Due to inherently higher scores for certain actions, the model shows a clear bias toward them.
3. **Learning Outcome (學習成效)**: Longer episodes and higher average rewards indicate that PPO successfully learned a stable interaction strategy.

Conclusion (總結)

This experiment demonstrates how to build a **Reinforcement Learning Environment (強化學習環境)** from **Simulated User Actions (模擬使用者行為)** and train it using **PPO**. The results show that the agent can quickly learn and converge to a high-reward strategy, proving that RL has potential to optimize user behavior patterns in gamification systems.

👉 This report now includes bilingual technical terms for clarity and professionalism.
Would you like me to add a **Future Work (未來改進方向)** section to suggest improvements for

reward shaping or integrating real user data? I’d be happy to help expand it.