TEAM NUMBER - 7
Name - Ria M Parikh, Nityashree, Rakshit V Agarwal ,Pranav P Pillai
SRN - PES1UG23AM234 , PES1UG23AM196 ,PES1UG23AM230 , PES1UG23AM208
Section - D

Analysis Report — Hangman

1. Key Observations

The most challenging aspect of this project was integrating **probabilistic reasoning (HMM)** with **reinforcement learning (DQN)**.
While the HMM provided strong prior probabilities for letters based on frequency and position, the DQN needed to **learn optimal guessing sequences** from limited and delayed rewards.
Handling **variable word lengths** was another major challenge as we had to pad state representations for stable training.

Reward scaling and epsilon decay tuning were crucial for convergence as too much exploration led to random guessing, while too little led to early stagnation.
Despite these hurdles, the hybrid model **significantly outperformed** simple frequency-based or standalone RL agents in terms of generalization and success rate.

2. Strategies

The **HMM** was designed using bigram-level modeling, capturing both **transition** and **emission probabilities** from a large English corpus.
These probabilities estimated the likelihood of each letter appearing at a specific position, guiding the DQN during exploration.

For the **DQN**, the state vector encoded:

- Current masked word pattern (padded to 15 letters)
- Guessed letters
- Remaining lives
- Progress ratio

Rewards were carefully shaped:

- **+0.2** for a correct guess
- **-0.3** for a wrong guess
- **+1.0** for successfully guessing the word

This smoother reward gradient helped faster convergence compared to sparse binary rewards.

3. Exploration vs. Exploitation

We adopted an **epsilon-greedy** policy starting with $\varepsilon = 1.0$ and decaying to 0.05 over time. This allowed for wide exploration early on, ensuring the agent experienced a variety of states before exploiting high-value actions later.

To make exploration more structured, we combined:

> **70% DQN predictions + 30% HMM probabilities**

This hybrid decision-making maintained a balance between **learned patterns** and **linguistic priors**, leading to improved stability and accuracy.

4. Future Improvements

Given more time, several enhancements could boost performance:

1. **RNN or Transformer encoders** to model sequential dependencies and context-aware guessing.
2. **Curriculum learning** — start with short, simple words and gradually move to longer, more complex ones.
3. **Prioritized experience replay** to emphasize rare or high-value learning situations.
4. **Dynamic reward scaling** to adapt based on word difficulty or length.
5. **Self-play or adversarial training**, where the agent challenges itself with generated words to improve robustness.