Answers to Analysis report questions:

- Key Observations: What were the most challenging parts? What insights did you gain?

The most challenging parts were handling the combinatorial explosion of word patterns and sparse reward signals in reinforcement learning. It was difficult to balance statistical modeling (HMM) with dynamic decision-making (RL) because both had different strengths.

We discovered that short words (2–4 letters) were surprisingly difficult due to limited contextual clues, while the HMM excelled at medium-length words (6–10 letters) where letter-transition regularities became more evident.

A major insight was that no single model generalized well to all word lengths or structures. The HMM captured language-level regularities, while the RL agent developed strategic memory, learning to prioritize vowels early and preserve lives in high-entropy states.

Another key finding was that the hybrid combination was essential — the dynamic weighting mechanism between probabilistic (HMM) and learned (RL) reasoning allowed the system to adapt its confidence based on uncertainty, leading to a more human-like guessing strategy.

- Strategies: Discuss your HMM design choices. Detail your RL state and reward design and why you chose them.

For the HMM, we designed length-specific models with Laplace (add-α) smoothing to handle unseen transitions and ensure numerical stability. Each HMM captured distinct bigram patterns, as letter transitions differ by word length (e.g., short words favor simpler transitions like “to”, while longer words exhibit richer dependencies).
We implemented a Forward–Backward algorithm to compute posterior probabilities efficiently given partial masks, ensuring the oracle could adaptively refine its predictions as new letters were revealed.

For the RL component, the state vector (~100 dimensions) encoded:

- One-hot representation of the mask
- Remaining lives
- Guessed letter indicators
- The full 26-dimensional HMM probability distribution

The reward design balanced immediate accuracy with long-term objectives (+1 for winning, −1 for losing), and included HMM confidence bonuses to reward consistency with linguistic

intuition.
This combination encouraged the agent to blend statistical reasoning with learned exploration, rather than relying purely on randomness.

Design insight: By structuring the reward around both “efficiency” (few wrong guesses) and “accuracy”, we aligned the learning process directly with the hackathon’s scoring formula.

- Exploration: How did you manage the exploration vs. exploitation trade-off?

Exploration–exploitation was handled using a multi-stage ε-greedy decay strategy combined with HMM-guided exploration.
Initially, ε = 1.0 encouraged full exploration; it decayed linearly to 0.1, promoting exploitation once the agent had a reliable Q-table.

During exploration, 70% of exploratory choices were biased by HMM letter probabilities (linguistic exploration), and 30% were purely random to preserve diversity.

We also integrated curriculum learning — starting with frequent/common words before moving to rare or longer ones — which stabilized learning early.
For the DQN variant, experience replay with priority sampling ensured that the model revisited states that yielded large reward gradients (like near-win or near-loss states).

Overall, the system learned to explore intelligently, guided by the language model rather than uniform randomness, leading to much faster convergence.

-Future Improvements: If you had another week, what would you do to improve your agent?

Given another week, we would:

1. Upgrade the HMM oracle to a Transformer-based sequence model (e.g., small BERT) to capture long-range letter dependencies beyond the Markov assumption.

2. Introduce hierarchical RL, where sub-agents specialize in different guessing strategies (e.g., vowel-first, prefix-based).

3. Expand the state representation to include morphological and semantic embeddings, allowing the agent to reason about word families (“play”, “player”, “playing”).

4. Employ multi-armed bandit algorithms for adaptive ε and learning-rate tuning during training.

5. Add adversarial evaluation: train the agent against increasingly difficult word sets (rare, low-frequency, or foreign-origin words) to improve robustness.

6. Implement transfer learning from general English text models (e.g., GPT-like embeddings) to pre-initialize language priors and reduce cold-start training time.

With these upgrades, the agent could approach near-human performance, combining statistical language intuition with adaptive strategic decision-making.

Final Results Summary (Hybrid: Improved HMM + CFP)

Metric	Meaning	Final Value
Top-1 Success Rate	% of times the correct word was the first guess	0.566 (≈ 56.6 %)
Top-3 Success Rate	% of times correct word was in top 3 guesses	0.853 (≈ 85.3 %)
Top-5 Success Rate	% of times correct word was in top 5 guesses	0.935 (≈ 93.5 %)
Win Rate	% of Hangman games completely won	0.356 (≈ 35.6 %)
Avg. Wrong Guesses	Average wrong letters per game	≈ 5.07
Avg. Repeated Guesses	Average repeated letters per game	≈ 0.00

Proxy Final Score	Composite performance metric (higher = better)	≈ −11,968