A Hybrid Ensemble of Specialized Experts for Solving Hangman

1. Introduction: From a Single Model to a Multi-Expert System

The core challenge in Hangman is to develop an agent that can navigate a problem of shifting complexity. A game begins with high ambiguity and concludes with a need for high-stakes precision. My initial approach involved a singular Bi-Directional LSTM (BiLSTM) model. The hypothesis was that its ability to process sequences bidirectionally would be sufficient to learn the contextual patterns of English words. While this model outperformed basic frequency methods, it lacked the specialized accuracy required for consistent success, particularly in the critical final stages of the game.

This led me to a key insight: **a single, generalist model is insufficient for mastering Hangman.** The optimal strategy changes as the game progresses. This realization formed the foundation of my final design: a dynamic, hybrid ensemble of five specialized neural network "experts," orchestrated by an intelligent inference engine.

2. Core Strategy: An Ensemble of Specialists

The strength of my approach lies in the principle of **specialization**. I developed five distinct models, each trained to excel in a specific phase of the game. This allows the agent to adapt its reasoning as a word is revealed, applying the most suitable logic for the current level of ambiguity. These experts are divided into two families: custom-built Transformers and fine-tuned CANINE models.

2.1. The Transformer Experts: Masters of Positional Context

I built a lightweight, character-level Transformer architecture from scratch. The self-attention mechanism is ideal for Hangman as it allows the model to weigh the importance of other characters when predicting a blank, based on their relative positions. I trained three distinct specialists:

- **Transformer Early (4-6 masks):** My specialist for high-ambiguity scenarios. Trained on words with many missing letters, it learns to identify broad structural

patterns and common letter groupings.
- **Transformer Mid (2-4 masks):** The mid-game workhorse. Trained on partially revealed words, it excels at using known letters as strong contextual clues to infer the remaining blanks.
- **Transformer Late (1-2 masks):** The precision expert. By training exclusively on nearly-complete words, this model is highly adept at resolving the final, often difficult, blanks where a single mistake results in a loss.

2.2. The CANINE Experts: Leveraging Pre-Trained Language Knowledge

To complement the positional expertise of the Transformers, I integrated Google's CANINE model, a powerful pre-trained language model with a key advantage for this task.

- **Why CANINE?** Unlike many models that see words as pieces (e.g., "hangman" as "hang" + "##man"), CANINE operates directly on characters. This provides a foundational, granular understanding of language that is perfectly suited to a character-based game.

To specialize this general knowledge, I performed Supervised Fine-Tuning (SFT) using **Low-Rank Adaptation (LoRA)**, an efficient technique to adapt large models without retraining them entirely. This produced two more experts:

- **CANINE Early (45%-80% masking):** Fine-tuned on heavily masked words, this model uses its vast pre-trained knowledge to make robust, high-probability guesses when the game state is highly uncertain.
- **CANINE Late (10%-40% masking):** Fine-tuned on words with fewer blanks, this expert is skilled at applying its deep language understanding to solve mid-to-late game scenarios with high accuracy.

3. The Inference Engine: A Dynamic, Multi-Stage Guessing Strategy

The final piece is the guess function, which acts as an intelligent engine to orchestrate the five experts. It follows a logical, multi-stage process at every turn.

- **Step 1: State Analysis & Dictionary Filtering:** The function first calculates the mask_ratio (the percentage of unknown letters) and filters a master dictionary list to create a smaller list of all possible words that fit the current pattern.
- **Step 2: Dictionary-First Heuristic (Early Game):** If the mask_ratio is very high (e.g., >70%), indicating the start of a game, the agent prioritizes a statistical approach. It calculates the frequency of un-guessed letters within the filtered dictionary and guesses the most common one. This provides a reliable and logical first move when the neural models have little context.

- **Step 3: Dynamic Expert Blending (Mid-to-Late Game):** Once a few letters are revealed, the agent transitions to its core neural strategy. Based on the mask_ratio, it selects a weighted combination of the most relevant experts.
    - *Mid-Game Example:* It might blend the predictions of canine_late (50% weight) and transformer_mid (50% weight).
    - *Late-Game Example:* It might favor precision, blending transformer_late (65% weight) and canine_late (35% weight).
- **Step 4: Blended Voting Mechanism:** The agent gathers the predictions (logits) from the selected models. For each blank space, it computes a blended probability score for all 26 letters by taking the weighted average of the experts' outputs. Each blank then "votes" for its top 3 most likely letters.
- **Step 5: Final Guess & Fallbacks:** The votes are tallied, and the letter with the most votes is selected as the guess. If this process yields no clear winner, the agent has robust fallbacks, reverting first to the dictionary frequency and finally to a general English letter frequency list to ensure a logical guess is always made.

4. Conclusion

My final strategy is a comprehensive solution that addresses the shifting complexities of Hangman. By creating an ensemble of specialized experts and orchestrating them with a dynamic inference engine, the agent can apply the most appropriate form of reasoning—be it statistical, positional, or deep linguistic—at every stage of the game. This multi-faceted approach overcomes the limitations of a single model, resulting in a highly adaptive and accurate Hangman player.