**1. Overview and Objectives**

**Objective:**  
Develop an efficient experimental framework to improve the model’s validation F1 score by systematically varying dataset formatting, augmentation strategies, and model architecture components. The design must address long per-epoch training times (≈30 minutes) and ensure fair comparisons, especially when dataset augmentation increases training time. Key performance metrics will be logged and compared using total training time (or equivalent gradient steps) as the normalization factor.

**Key Deliverables:**

* A **CSV log** that records each trial’s unique identifier and parameter settings.
* A **JSON record** per trial that includes:
  + The learning curve dataframe (tracking training and validation metrics per epoch).
  + The peak validation F1 score and the epoch at which it occurs.
  + Detailed trial parameters and any additional observations (e.g., from layer visualizations).
* A **visualization tool** (neuron “viewer”/heatmap) to assess layer sizes.

**2. Experimental Design Approach**

**A. Statistical Experiment Design with Fractional Factorial Principles**

Given the many factors involved, we adopt a **fractional factorial design** to efficiently screen the main effects and selected interactions. We recommend a two-stage process:

1. **Screening Phase:**
   * **Goal:** Identify which factors most significantly affect F1 score.
   * **Design:** Start with 10 trials (initially using a fractional factorial design) focused on dataset formatting and baseline model architecture for screening.
   * **Note:** Although 10 trials might be sufficient for screening, the expectation is to add more trials if significant interactions emerge.
2. **Optimization Phase:**
   * **Goal:** Fine-tune promising parameters and explore interactions in more detail.
   * **Design:** Expand the experiment based on screening outcomes, possibly adding more runs or using response surface methodology (RSM) around promising settings.

**B. Factors and Levels**

**1. Dataset Format Factors:**

* **Smoothing Type:**
  + Custom smoothing
  + Binary (no smoothing)
  + Binary label smoothing
* **Background Column:**
  + Included
  + Removed
* **Early Rewarding (for custom smoothing only):**
  + Enabled
  + Disabled

**2. Dataset Augmentation Factors:**

* **Augmentation Factor:**
  + Baseline (1x)
  + Moderate (e.g., 5x)
  + Maximum (10x)
* **Window Shift Strategy:**
  + Incremental (one window shift at a time)
  + Highly mixed shifts

**3. Model Architecture Factors:**

* **Dilation Setting:**
  + Standard
  + Less/Slower
* **Attention Layer Placement:**
  + No attention layer
  + Masked attention at the beginning
  + Pooling + attention at the end of convolution
  + Attention after concatenation
* **Regularization (L1 and L2) on Neurons:**
  + Absent
  + Present

**C. Hyperparameter Tuning**

To prevent confounding from hyperparameter variability:

* **Decoupled Tuning:**
  + **Stage 1:** Perform initial hyperparameter tuning (e.g., learning rate, batch size) using Keras Tuner
  + **Stage 2:** Fix these hyperparameters when running the factorial design experiments.
* **Option to Integrate:**
  + If certain hyperparameters are expected to interact with the primary factors, they can be included in an expanded design—but this will increase the number of trials. A nested or sequential approach is preferred to keep the design manageable.

**3. Experiment Timeline and Ordering**

**Stage 1: Establish a Baseline and Screen Factors**

* **Baseline Trial:**
  + Run the current best-known configuration and log all details.
* **Screening Trials:**
  + Use a fractional factorial design (starting with roughly 10–12 trials) varying dataset formatting factors (smoothing, background inclusion, early rewarding) and standard architecture settings.
  + Keep augmentation at baseline (1x) and model architecture at standard dilation with no added attention layers.
  + Log each trial in the CSV and JSON formats.

**Stage 2: Augmentation and Architecture Experiments**

* **Dataset Augmentation:**
  + With the best dataset format identified from screening, vary the augmentation factor and window shift strategy.
  + Normalize performance comparisons using **total training time** (or number of gradient steps) rather than epochs, as increased augmentation directly impacts epoch duration.
* **Model Architecture Modifications:**
  + With the best dataset format and augmentation settings, test architectural changes:
    - Begin with variations in dilation.
    - Sequentially introduce different attention layers (masked at the input, pooling + attention at the convolution end, and post-concatenation attention).
    - Explore the effect of neuron regularization.
  + Consider running some experiments in parallel if compute resources allow.

**Stage 3: Cross-Validation and Final Optimization**

* **Combine Promising Configurations:**
  + Run a set of final trials that combine the best dataset format, augmentation strategy, and architecture modifications.
  + Use performance per unit of training time as the key metric.
  + Perform ANOVA or regression analysis on the aggregated data to verify statistically significant factors and interactions.

**4. CSV and JSON Logging Structure**

**CSV File Structure**

Each trial record should include the following columns:

* **Trial Number:** Unique identifier.
* **Dataset Format:**
  + Smoothing Type
  + Background Column (Included/Removed)
  + Early Rewarding (Enabled/Disabled, if applicable)
* **Dataset Augmentation:**
  + Augmentation Factor (e.g., 1x, 5x, 10x)
  + Window Shift Strategy (Incremental/Highly Mixed)
* **Model Architecture:**
  + Dilation Setting (Standard/Less-Slower)
  + Attention Layer(s) (None, Masked, Pooling + Attention, Post-Concatenation)
  + Regularization (Yes/No)
* **Fixed Hyperparameters:** (as determined in the tuning phase)
* **Total Training Time:** (or total gradient steps)
* **Notes/Observations:** Additional observations or unexpected behaviors.

**JSON Record for Each Trial**

For each trial (indexed by trial number), save a JSON file containing:

* **Trial Info:** All parameters corresponding to the CSV entry.
* **Learning Curve DataFrame:** Training and validation metrics logged per epoch.
* **Peak Validation F1 Score:** The highest F1 score achieved.
* **Epoch of Peak:** The epoch number at which the peak occurred.
* **Total Training Time:** Normalized training duration.
* **Additional Notes:** Any insights (including visualization outputs) that may inform further tuning.

**5. Visualization: Neuron “Viewer” / Heatmap**

* **Purpose:**  
  Develop a visualization tool to view neuron weights, helping identify potential issues such as oversized layers.
* **Implementation:**
  + Create heatmaps that highlight neurons with near 0 weight. Consider reducing filter count if there are many.
  + Record any notable observations from these visualizations in the JSON logs.
* **Usage:**  
  Use these visualizations to refine model architecture choices in subsequent experiments.

**6. Summary and Best Practices**

* **Statistical Design:**
  + Begin with a fractional factorial design (starting with 10 trials for screening) and extend based on initial results. Sequential experimentation allows for refinement without overwhelming resource demands.
* **Decoupled Hyperparameter Tuning:**
  + Tune hyperparameters separately to establish a robust baseline before introducing additional factors.
* **Fair Performance Comparison:**
  + Use total training time (or gradient steps) rather than epoch counts for comparing experiments with varying dataset sizes.
* **Parallelization and Iteration:**
  + Run independent trials in parallel where possible to shorten the overall experiment timeline.
* **Detailed Logging:**
  + Maintain comprehensive CSV and JSON logs to track configurations and performance outcomes. This systematic logging is crucial given the long training times and complexity of the design.