Plan: EfficientNet-Lite0 + TRM Fusion with M2N2 Optimization

Student Name: Muhammad Qasim Jalil
Student ID: 539433
Section: RAI - 2025

1. Executive Summary

This project aims to implement a hybrid computer vision architecture fusing **EfficientNet-Lite0** (backbone) with **TRM** (Tiny Recursive Model) to enhance feature reasoning. To maximize performance and stability, we will employ **M2N2** (Model Merging of Natural Niches), an evolutionary weight-merging algorithm, to fuse the best traits of multiple trained hybrid models.

The provided `M2N2.pdf` (Sakana AI, 2025) describes an *evolutionary model merging algorithm*. As per the instruction to use the provided PDFs, **we will implement M2N2 as the weight-merging algorithm described in `M2N2.pdf`**, applying it to merge checkpoints of the hybrid model to boost accuracy.

2. Resources & Constraints

- **Hardware:** Nvidia MX450 (2GB VRAM), 16GB RAM, i7 CPU.
- **Constraints:**
    - Strict VRAM limit requires small batch sizes (32-64) and reduced model width.
    - Training from scratch on CIFAR-10.
    - Limited training budget (approx. 50 epochs suggested).

3. Architecture Design

3.1 Backbone: EfficientNet-Lite0 (Modified)

- **Source:** Adapted from EfficientNet principles (Tan & Le, 2019) but simplified ("Lite") as per PRD.
- **Modifications for CIFAR-10 (32x32 input):**
    - **Stride Adaptation:** The first convolution stride will be set to 1 (instead of 2) to preserve spatial dimensions.
    - **Stem:** 3x3 Conv, 32 channels.

    - **Blocks:** MBConv blocks (inverted residuals with SE removed for "Lite" compliance).
    - **Output:** Feature map of size `(Batch, 1280, 1, 1)` (Global Average Pooled).

3.2 Reasoning Module: TRM (Tiny Recursive Model)

- **Source:** `TRM.pdf` (Jolicoeur-Martineau, 2025).
- **Placement:** Inserted after the final convolutional features of EfficientNet, before the classifier head.
- **Adaptation:**
    - The paper's TRM uses ~7M params. Given the 2GB VRAM limit, we will implement a **"Nano-TRM"**:
        - Reduced embedding dimension (*D=64 or 128* instead of standard).
        - Recursion depth (*n=2, T=2*) to save compute.
    - **Input:** Flattened features from EfficientNet.
    - **Function:** Recursively refines the feature vector *z* (latent reasoning) and *y* (answer/logit candidate).

3.3 Fusion Algorithm: M2N2 (Model Merging)

- **Source:** `M2N2.pdf` (Abrantes et al., 2025).
- **Implementation:**
    - We will train **2 distinct instances** (Seeds A and B) of the `EfficientNet+TRM` hybrid model.
    - **Merging:** Apply the M2N2 algorithm (Evolutionary search with SLERP and Attraction) to merge the weights of Seed A and Seed B.
    - **Goal:** Produce a final "Fused" model that outperforms the individual seeds.

4. Implementation Steps

Phase 1: Data & Utilities

- `src/data/cifar10.py`:
    - Loaders for CIFAR-10.
    - Transforms: RandomCrop(32, padding=4), RandomHorizontalFlip, Normalize.
- `src/utils/seed.py`: Deterministic seeding.

Phase 2: Model Architecture

- `src/models/efficientnet_lite.py`:

    - `MBConvBlock` class.
    - `EfficientNetLite0` class with adjusted strides.
- `src/models/trm.py`:
    - `TRMBlock`: Implements Algorithm 3 from `TRM.pdf`.
- `src/models/hybrid_model.py`:
    - Combines EfficientNet backbone + TRM module + Classifier.

Phase 3: M2N2 Merging Engine

- `src/m2n2_fusion.py`:
    - `ModelArchive`: Stores model weights and scores.
    - `merge_models(model_a, model_b)`: Implements the SLERP fusion with split points (Eq 2 in PDF).
    - `evolve()`: Runs the evolutionary loop to find best merge parameters.

Phase 4: Training & Experiments

- `configs/train.yaml`:
    - Batch size: 64
    - LR: 0.001 (AdamW)
    - Epochs: 40 (Baseline), 40 (Hybrid)
- **Experiment A (Baseline):** Train `EfficientNet-Lite0` (x2 seeds).
- **Experiment B (Hybrid):** Train `EfficientNet-Lite0 + TRM` (x2 seeds).
- **Experiment C (Fusion):** Run `M2N2` on the weights from Experiment B to create the final model.

5. Evaluation & Deliverables

- **Metrics:** Top-1 Accuracy, Parameter Count, Inference Latency.
- **Results:**
    - Table comparing Baseline vs. Hybrid vs. Hybrid+M2N2.
    - T-test significance.

6. Risks & Mitigations

- **OOM (Out of Memory):** If 2GB VRAM is exceeded, reduce Batch Size to 16 and TRM dim to 64.
- **M2N2 Complexity:** The full evolutionary search might be slow. We will limit the "generations" for the merge search to a small number (e.g., 10) to demonstrate the concept within the timeline.