Feature Specification: Training UI & Thinking Model Integration

Status: Draft
Owner: Engineering
Date: 2026-01-05

1. Overview

This document outlines the changes required for the “Second Me” training interface, specifically the introduction of the **Thinking Model** configuration and the overhaul of the training parameter controls.

2. User Interface Changes

2.1 Training Configuration Card

The main entry point is a new `TrainingConfig` component. It must handle the following states: - **Idle:** Show “Start Training”. - **Training:** Show “Stop Training” with a spinner. - **Suspended:** Show “Resume Training”.

Key Requirements: - **Support Model:** Users must be able to select between OpenAI and Custom models for data synthesis. - **Base Model:** Dropdown selection for the base model (e.g., Llama 3, Mistral). - **Advanced Params:** Learning rate, Epochs, and

Concurrency must be adjustable but protected during active training.

2.2 The “Thinking Model” Toggle – NEED A PRACTICAL EXAMPLE

We are introducing a Chain-of-Thought (CoT) toggle. - **Constraint:** The toggle must be disabled if the `ThinkingModelConfig` is incomplete (missing API key or endpoint). - **UX:** If a user tries to enable CoT without config, show a warning pulse animation on the “Thinking Model” button.

```
// Logic for handling the Thinking Model toggle
<Checkbox
  checked={trainingParams.is_cot}
  disabled={disabledChangeParams}
  onChange={(e) => {
    e.stopPropagation();
    if (!thinkingConfigComplete) {
      setShowThinkingWarning(true); // Triggers red pulse animation
      return;
    }
    updateTrainingParams({ ...trainingParams, is_cot: e.target.checked
  }}
/>
```

3. Backend Integration

3.1 Data Synthesis

The backend needs to accept the new `data_synthesis_mode` parameter: * `Low`: Fast processing. * `Medium`: Balanced. * `High`: Rich synthesis (slower).

3.2 CUDA Acceleration

We need to detect system capabilities before enabling the checkbox. * If `cudaAvailable` is false, the checkbox is disabled and stylized as gray.

4. Open Questions / Review Items

1. **Validation:** Should we hard-limit the `learning_rate` to `0.005` on the backend as well?
2. **Resume Logic:** Does “Resume Training” preserve the *exact* epoch state, or does it restart the current epoch?
3. **Visuals:** Review the `custom-message-success` styling below. Is the green border (`#b7eb8f`) accessible?

```
/* Proposed success message styling */
.custom-message-success .ant-message-notice-content {
    background-color: #f6ffed;
    border: 1px solid #b7eb8f;
    padding: 0.5rem;
    border-radius: 0.25rem;
}
```

5. Next Steps

- [x] ~~Review this spec on the Supernote~~.
- [x] ~~Annotate~~ UI concerns directly on the code blocks.
- [x] ~~Finalize the~~ `ThinkingModelModal` layout.

Done