Machine Learning Powered Level-Matching Development Plan

Best Model Choice

LightGBM Ranker (pairwise ranking) + hard physics masks + Hungarian assignment.

Why: Tree boosting handles nonlinear feature interactions, missing values, mixed discrete and continuous features, and trains fast on small labels. Ranking fits the natural task: for each level in one scheme, rank candidate matches in the other scheme, then enforce one-to-one globally.

End-to-End Pipeline

1. Per-Scheme Affine Calibration

Fit robust $\mathbf{E' = aE + b}$ per dataset against a reference using RANSAC on coarse nearest neighbors. Use residuals in features. Keep **a** and **b** as per-scheme metadata.

2. Candidate Generation

For each level **A_i**, collect **B_j** with $\mathbf{|E'_A - E_B| \le w}$ and $\mathbf{z \le z_{max}}$ using $\mathbf{\sigma_{ij} = \sqrt{\sigma_A^2 + \sigma_B^2}}$. Typical **w = 10 to 20 keV**, **z_max = 4**.

3. Physics Masks

Remove candidates that are hard-forbidden by **L** or **Jπ** rules or by reaction selectivity. Keep a soft prior feature for **L** or **Jπ** compatibility even when not forbidden.

4. Features per Pair (A_i, B_j)

- **Core:** z, |ΔE|, sign(ΔE)
- **Quantum numbers:** trinary L match, trinary Jπ match, parity match, ΔJ
- **Population priors:** experiment channel flags, known selectivity, beam-target metadata
- **Spectroscopy patterns:** γ-out Jaccard on binned energies, top-k line overlap, intensity similarity, branching vector cosine
- **Structure context:** neighbor spacing similarity before and after, local level density around E, difference in cumulative counts
- **Calibration context:** a and b residuals, per-dataset energy scale uncertainty

Missing values are fine. LightGBM handles them.

5. Learning Objective

Train **LightGBM Ranker** with pairwise objective. Group by query = A index. Positives are true matches, negatives are other candidates for the same A. If labels are few, start by heuristics for pseudo-labels, then iterate.

6. Scoring and Assignment

Predict scores **s_ij**. Convert to costs **C_ij = −log(s_ij+ε)**. Keep only candidates that pass hard masks and **z ≤ z_max**. Solve one-to-one with **Hungarian**. If you need monotonicity in energy, use a dynamic-programming matcher with i increasing implies j increasing.

7. Unmatched Handling

Leave **A_i** or **B_j** unmatched if the best score < threshold or cost > cutoff. Report top-k alternates per **A_i** for curator review.

How to Use with Your Datasets

1. Use your first dataset as reference. Fit affine **a**, **b** for each of the other 9 datasets.
2. Generate candidates between the reference and each dataset using your **E** and **σE**, and your **Jπ** and **L** annotations.
3. Start training with a few manually confirmed matches across energy regions. Add hard negatives where **L** or **Jπ** is disallowed.
4. Run prediction to get a ranked match list with top-1 assignment plus top-3 alternates for review.

Why Not Deep Nets

Tabular, small-to-medium labels, physics rules, and need for interpretability favor gradient-boosted trees and ranking objectives. They are accurate with minimal tuning and give SHAP-style feature attributions for sanity checks.