8) Evaluation Protocol Report（評估流程報告）

Date: 2025-12-24

Goal

Define a consistent evaluation protocol for teachers, ensembles, and students.

Datasets / splits

Common evaluation sources used in this project:

- HQ training manifest evaluation:
  - `Training_data_cleaned/classification_manifest_hq_train.csv`
- Mixed-source robustness test:
  - `Training_data_cleaned/test_all_sources.csv`
- RAF-derived benchmarks:
  - `Training_data_cleaned/rafdb_basic_only.csv`
  - additional RAF compound/RAF-ML CSVs under `Training_data_cleaned/`

Metrics reported

Per run, store metrics in JSON files under the run folder:

- Classification metrics:
  - Accuracy
  - Macro-F1
  - Per-class F1
- Reliability / calibration metrics:
  - Negative Log-Likelihood (NLL)
  - Expected Calibration Error (ECE)
  - Brier score
  - Temperature scaling results (global temperature, post-calibration NLL/ECE)

Where metrics come from

- Reliability computation helper:
  - `scripts/compute_reliability.py`
- Live/demo scoring helper (if used for demo logs):
  - `scripts/score_live_results.py`

Temperature scaling

- Apply post-hoc temperature scaling on validation logits to reduce miscalibration.
- Report both raw and temperature-scaled NLL/ECE.

Ensemble evaluation

- Evaluate ensembles in logit space:
  - weighted logit fusion
  - best weights selected from a target benchmark (e.g., `test_all_sources.csv`)

Output artifacts per evaluation

Expected in a complete run folder:

- `history.json`
- `reliabilitymetrics.json`
- `calibration.json`
- `best.pt`

Next steps

- Standardize one command/script to evaluate any checkpoint on a chosen manifest and always emit the same JSON schema.