# General Data Preparation Pipeline

February 16, 2026

## 1 Shared Preparation (Supports Ideas 1–4)

To move quickly while keeping all four directions feasible, we first implement a shared data pipeline that supports: (i) variable-length context slicing, (ii) montage/channel variability, (iii) missingness masks, (iv) stress-test perturbations, and (v) optional embedding caches (e.g., SleepFM). This maximizes overlap and minimizes rework when deciding which direction to pursue.

### 1.1 S0. Storage Layout on CC (Realistic and Scalable)

Use a persistent high-quota location for derived artifacts and a fast temporary location for intermediate caches:

- `$PROJECT/psg/{dataset}/raw/` (symlink or extracted official structure)
- `$PROJECT/psg/{dataset}/derived/` (preprocessed signals, embeddings, metadata)
- `$PROJECT/psg/unified/metadata/` (global tables and splits)
- `$SCRATCH/psg_cache/` (temporary caches, shards, memmaps)

Avoid storing large binary artifacts in `$HOME`.

### 1.2 S1. Unified Metadata Table (Main Accelerator)

Create a single `csv/parquet` that indexes every subject-night across STAGES/SHHS/APPLES/MrOS. For each record store:

- dataset name, subject ID, night ID, recording duration, timestamps
- available channels list and mapping to modality families (EEG/EOG/EMG/ECG/Resp)
- sampling rates per channel (or per modality family)
- label availability flags and label paths (staging per 30s, night-level labels)
- train/val/test split assignment (subject-wise) and fold ID
- optional "site/cohort" ID if available (useful for shift evaluation)

This table is reused by all ideas.

### 1.3 S2. Standardize Preprocessing (Minimal, Consistent First)

Implement a consistent baseline preprocessing per night:

- resampling to a common rate per modality family *or* keep native rates with on-the-fly resampling
- light bandpass filtering (initial defaults) and robust per-night channel scaling (z-score or median/IQR)
- store per-window validity masks (invalid if missing/corrupted)

Cache the preprocessed per-night arrays once, then train from cached arrays.

### 1.4 S3. Create a Windowed Cache (Enables Variable Lengths and Fast Training)

Define canonical units:

- epoch = 30s (for staging)
- chunk = 2–5 minutes (for exits/long-context modules)
- night windows = {start, middle, end, full} (for optional coverage experiments)

For each night, cache:

- signals (or embeddings) per modality family over time
- staging labels aligned to 30s epochs (if available)
- night-level labels (AHI/apnea/etc.) with availability flags
- modality-family masks $\mathbf{z}_m[t]$ indicating which modalities are present/valid per time step

### 1.5 S4. Splits that Support Both In-domain and Domain Shift

Store two complementary split regimes:

- in-domain: subject-wise split within each dataset (avoid leakage)
- cross-dataset: train on dataset A, test on dataset B (for shift stress tests)

### 1.6 S5. Optional: SleepFM Embedding Cache (Speed Booster Later)

If using SleepFM, cache embeddings per night:

- $\mathbf{H}_m[t]$ per modality family and time step
- store embedding time step size (e.g., 5s or 30s) to convert minutes $\leftrightarrow$ steps

You can implement the pipeline first on raw signals and later add embedding caching without changing the dataset index.

## 2 Idea-Specific Preparation (What Each Direction Needs in Addition)

### 2.1 Idea 1 (AMTA+CSL): Extra Preparation

This direction primarily requires efficient *random variable-length slicing* and paired short/long views.

- **I1. Anchored sampling for staging:** index valid anchor epochs $t$ with label $y_t$ so the loader can return past-only context ending at $t$.
- **I2. Length menu:** predefine a small length set (log-spaced), e.g., $\{2, 5, 10, 20, 40, 80\}$ minutes, and implement loader functions returning two views $(L_s, L_\ell)$ for the same anchor/sample.
- **I3. Optional coverage windows (night-level):** precompute indices for {start, middle, end, full} windows per night (nights vary in duration).

### 2.2 Idea 2 (Early-Exit Length Selection): Extra Preparation

This direction needs fixed checkpoints and prefix states, which still rely on the same slicing primitives.

- **I4. Checkpoints in model steps:** define exits as steps rather than minutes (e.g., if step=30s: 2/5/10/20/40 min $\rightarrow$ 4/10/20/40/80 steps).
- **I5. Longest-prefix availability:** ensure the loader can always provide the maximum context $L_K$ for the sampled anchor, while shorter prefixes are subsets of it.
- **I6. Optional coverage support (night-level):** enable prefix reading within a chosen window (start/middle/end/full) if later adding coverage selection.

### 2.3 Idea 3 (Safe TTA + Stress Suite): Extra Preparation

This direction is evaluation- and shift-oriented: it needs a reproducible perturbation suite and target streams.

- **I7. Stress-test transforms as dataset wrappers:** implement functions to apply missing modalities, montage perturbations (channel subsets), intermittent dropout, artifact injection, and sampling mismatches at signal-level or embedding-level.
- **I8. Target "streams" for adaptation:** store per-night sequential ordering (time index) so test-time adaptation can process unlabeled sequences in order.
- **I9. Prototype support (for margin gate):** after source training, compute and save class prototypes (mean embeddings per class) from the training set for use at test time.

### 2.4 Idea 4 (Modality-aware MoE Routing): Extra Preparation

This direction needs reliable modality masks and missingness augmentation schedules.

- **I10. Per-window modality-family masks:** ensure $\mathbf{z}_m[t]$ exists at the same temporal resolution as your model inputs (epoch or chunk).
- **I11. Missingness/montage augmentation schedule:** implement seeded training-time augmentation:
  - whole-modality drop (entire sample)
  - EEG channel subset sampling (montage simulation)
  - intermittent segment masking (contiguous gaps)
- **I12. Optional cheap reliability proxies:** if desired, precompute lightweight per-window statistics per modality (RMS/variance, clipping ratio, flatline ratio, high-frequency energy ratio) to help routing; this avoids needing dataset-provided quality scores.

## 3 Mutual Steps (Max Overlap Across Ideas 1–4)

The following steps are shared and can be executed before choosing a direction:

1. Unified metadata table (subjects/nights, labels, channel lists, modality mapping, splits).
2. Cached per-night standardized arrays and aligned labels (staging + night-level).
3. Efficient variable-length slicing interface by (night_id, anchor $t$, length $L$).
4. Modality-family masks per window and (optional) segment missingness masks.
5. Stress-test transform wrappers (even if initially unused).
6. Optional embedding cache (SleepFM) keyed by the same metadata indices.

# 4 Fastest Practical Start (Recommended)

To maximize speed:

- Start with one dataset that is easiest to parse on CC (often SHHS or STAGES depending on access/tooling).
- Implement staging first (many samples, quick feedback), then add one night-level label next.
- Once one dataset runs end-to-end, replicate the adapter for additional datasets using the same unified metadata format.